Annals of Operations Research

Solving a Large-Scale Integrated Fleet Assignment and Crew Pairing Problem

--Manuscript Draft--

Manuscript Number:	ANOR-D-14-00363R1
Full Title:	Solving a Large-Scale Integrated Fleet Assignment and Crew Pairing Problem
Article Type:	Original Research
Keywords:	Fleet assignment; Crew pairing; Airline operations planning; Large-scale neighborhood search
Corresponding Author:	Melda Ormeci Matoglu University of New Hampshire Durham, UNITED STATES
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	University of New Hampshire
Corresponding Author's Secondary Institution:	
First Author:	Okan Orsan Ozener
First Author Secondary Information:	
Order of Authors:	Okan Orsan Ozener
	Melda Ormeci Matoglu
	Gunes Erdogan
	Mohamed Haouari
	Hasan Sozer
Order of Authors Secondary Information:	
Funding Information:	TUBITAK (110M308) — Dr. Okan Orsan Ozener
Abstract:	Airline planning problems are typically decomposed into smaller problems, which are solved in a sequential manner, due to the complexity of the overall problems. This results in suboptimal solutions as well as feasibility issues in the consecutive phases. In this study, we address the integrated fleet assignment and crew pairing problem (IFACPP) of a European Airline. The specific network and cost structures allow us to develop novel approaches to this integrated problem. We propose an optimization-driven algorithm that can efficiently handle large scale instances of the IFACPP. We perform a computational study on monthly flight schedules to test the performance of our solution method. Based on the results on instances with up to 27,500 flight legs, we show that our algorithm provides solutions with significant cost savings over the sequential approach.

We like to express our sincere appreciation and thanks to the AE and Reviewers for their careful review of our paper. They have made very valuable suggestions and comments. We tried to address them to the best of our ability. We believe with these changes our motivation in studying this challenging problem and contributions have become clearer. The exposition has improved significantly, our solution approach is better explained. Overall we believe we have a stronger paper.

Below you will find our point by point responses to individual comments.

Reviewer #1: This paper focuses on a combination of two very important problems in airline planning process, namely, fleet assignment and crew pairing. While these problems have been extensively dealt with in the past literature (both separately and jointly), the size of the problem being tackled in this case study is significantly larger than most in the literature. Also, as rightly pointed out by the authors, there are a number of special features of the problem that make this an interesting enough and worthwhile problem to solve using novel approaches. The approach proposed by the authors certainly has several novel features and is as such of interest to be published in a top journal. The paper is very well written in general and the problem as well as the need for novel solution approaches is well-motivated. There are, however, a number of critical issues with the current version of paper that make it unsuitable for publication in its current form, in my opinion. Below I have listed some of the major and minor issues that need to be addressed.

Major Comments:

1) The biggest issue is the lack of a real benchmark to compare the methods against. The authors provide some benchmarks based on one type of methods (sequential) and compare the other two types of methods (iterative and integrated) against the sequential method. However, these are all methods proposed by the authors are do not validate in any sense how good or bad the proposed methods are. The three typical ways to circumvent this issues are a) comparison against the actual solution used by the airline, that is, the actual fleeting and crew scheduling decisions by the airline, OR b) comparison against the global optimal FP+CPP solution (this can be obtained by running an exact solution algorithm a very large amount of time) - also called optimality gap, OR c) showing a small gap when compared with the LP relaxation (or some other mind of relaxation of the problem), OR d) comparison against the established benchmark in literature, such as the best known method for solving these problems exactly, as described in literature. In the absence of any such comparison, the value of the approach developed by the authors is very much suspect. Addressing this comment will also require re-writing sections 5 and 6 accordingly.

We thank the reviewer for providing such a comprehensive comment with multiple possible approaches to address it. To gauge the quality of our solution we computed an upper bound. We implemented an upper bound using a multi-commodity network flow problem formulation for the Fleet Assignment component and ignored the crew pairing cost. We solved an LP relaxation of the multi-commodity network flow problem. Based on this the average deviations from the upper bound for the 5 instances is 8.1%. The deviations from the upper bound is as follows for the 5 instances:

Instance	Deviation from UB
1	7.89%
2	7.98%
3	8.10%
4	8.62%
5	8.15%

2) The authors have developed and described a number of algorithms including the sequential method, the iterative method, the integrated method, etc. for solving this problem. However, the decisions made throughout the paper are fairly arbitrary (or at least appear arbitrary). There are innumerable variants of these heuristic steps and their combinations that can be tried and tested and it is highly unclear why these are the once that were worthy of being tried or being reporting here. Similarly, there are many places where parameters are picked (like alpha, beta, n_sub, n_iter, n_l, n_it, and so on, on pages 17 and 18) seemingly arbitrarily. Clear justification is needed for all these decisions throughout the paper to make it clear to the reader that the decisions are not arbitrary but are made based on sound logic.

We have included a sensitivity study based on the values of the different parameters. As it can be seen the change in the objective function becomes very small after a number of iterations (less than 0.3%). We have provided also plots on the progression of the procedures as the number of iterations increase. There was a reference to parameter n_ell in the original manuscript, however that was a typo as there is no such n_ell parameter in any of the procedures.

3) On page 8 and elsewhere in the paper, the authors do a good job of identifying the special features of the network that they are solving (e.g., one crew base, one hub, etc.). However, an important attribute of good quality academic research is to generate insights that can be carried over to solving other problems beyond the exact problem instance being solved in that study. What are such major takeaways that this paper generates? What are the other problem contexts (beyond this one European airline) for which these same insights apply? I do believe that the problem being solved by the authors is not so specific that it has no generalizability. But currently there is no evidence of that being provided. This makes the paper look like an exercise in solving a very specific problem instance with no generalizability to other instances beyond this particular European airline.

The particular variant of the Fleet assignment and Crew Pairing problem we study in this paper arises in many airlines, in particular in airlines where crews are paid based on a fixed salary. The aperiodicity of the flight schedule, a feature common in many European airlines, requires addressing the monthly problem, which makes the problem size significantly big. We show that by taking into account the network structure, in this case a single base with many point to point flights, we are able to generate good solutions in reasonable time. The particular EAC problem allows us to demonstrate our approach in a real-world setting. We added comments in the introduction and conclusion section to emphasize the generalizability of our approach.

4) Many of the heuristic steps such as FP Initialization, FP local search, etc. need to be described accurately and need to be substantially re-written in mathematically accurate ways. For example, in FP Initialization on page 10, several claims are made, which are not necessarily correct or at least not obvious. E.g. a) Chains are said to be concatenated to reduce to number of available aircraft. Is this always possible? If so, why and if not, what to do then?, b) The same paragraph then says, "Here we lift some of the restrictions and include the possibility of repositioning". This is very vague description for an OR journal. How does this "lifting of some restrictions" work? What does that mean? c) Do the chains have to begin and end at the same airport? If so, how is that ensured? If not, why not? d) How are profits for chains computed? Please carefully rewrite the heuristic descriptions to make sure that these and similar loose ends are tied in the descriptions of all the heuristic and exact procedures.

The respective paragraph in the manuscript is modified as follows:

> We generate these chains in a greedy manner, that is, after adding the leg which arrives at time ta to the chain, we obtain the “ready time" as ta +r, where r is the minimum turn time, **which is 30 minutes.** Next, we identify the first fight leg that departs from that particular airport at or after the ready time and add that flight leg to the chain and we continue in this manner. **If there is no flight departing from that particular airport up until a certain time (14 hours for domestic and 48 hours for international locations), then there are no more feasible legs to be added to this chain** and we start a new chain from the first unassigned leg in the time sorted leg list. Once all legs are included to a chain, we concatenate feasible chains, to reduce the number of chains to the number of available aircraft. This concatenation step is different than adding legs to a chain in a greedy manner. **Here we include the possibility of repositioning of an aircraft and concatenate two previously generated chains if one aircraft can feasibly perform one after the other while accounting for the deadheading flight time from the arrival location of the last flight of the first chain to the departure location of the first flight of the second chain. As in the previous routine, concatenation process is also greedy hence a chain is concatenated to the earliest feasible chain. Note that the chains are not required to start and terminate at the same airport as the initial condition (locations of aircrafts, crews, etc.) at the beginning of each month might be different from one another.**

We also modified the description of FP Initialization and added the paragraph below to the manuscript:

> **Note that this procedure does not guarantee that the number of chains at the end is less than or equal to the number of available aircraft. However it is also important to note that in flight network construction stage airlines do take the number of aircraft information into account and decide on the flights based that information. In that aspect, the flights to be flown by the airlines has a certain structure and the respective data does not consist of arbitrary flight legs from arbitrary locations to some other arbitrary locations. In fact, the flights schedules have certain pattern, which makes the aircraft assignment process relatively easier.**

5) Procedure FP_LS needs a significant rewrite. Several issues can be spotted. a) I think the line starting with "Identify the set" needs to be inside the For loop not outside, I believe. b) There is a terminology issue with the way you define a 'switch point' in P_ij. Is a 'switch point' supposed to be a single flight or a pair of flights? For example, in the example on page 11 bottom, are 3 and d two different switch points or

one? These terms need to be carefully defined before being used. c) Can same i and j have more than one switch points? d) How are C_tilda[i], C_tilda[j], and n_sub defined? e) Replace the words "each" on lines 12 and 14 with "either" because there are only 2.

The description of the algorithm is updated. We have also modified the reference to the switch point and iteration limit as follows:

> "We illustrate this with a small example. Suppose there are two chains, 1-2-3-4-5-6, and a-b-c-d-e-f-g-h, with sub-chains 4-5-6 and e-f-g-h. **Then, the pair $(3,d)$ is a potential switching point.** Then we create new chains by exchanging these sub-chains within the long chains to create new chains. In this small example, the new chains would be 1-2-3-e-f-g-h and a-b-c-d-4-5-6. **Note that a pair of chains might have several switch points depending their** compatibility…**Repeating this step for a certain number iterations (n_sub) quickly brings the problem to a ``better" fleet assignment solution compared to the initial solution of FP_0.** The details of this procedure are as follows:"

Note that we also carried out a sensitivity analysis on the value of the parameters such as n_sub, showing that the performance of our algorithm is robust with respect to different values of this parameter.

6) Similarly, Procedure FP_ILS needs to be clearly rewritten. Most of it is currently just a copy of FP_LS and the difference between the two is highly unclear. Page 12 lines 29 through 31 say something about the leg basis versus mini chains. That is really unclear. Also, how is n_iter decided? Page 13, line 9 talks about forcing a switch (by the way, please explain this step more carefully since this sems to be the only difference between FP_LS and FP_ILS). What does that mean? How is that done? Please avoid these various sources of vagueness and rewrite this procedure clearly.

The description of the algorithm is updated. We have also modified the reference to the leg based procedure as

> "At this point, we have a "good" solution to the FP, and we start a more detailed iterated local search. In essence, as a subroutine we use a local search algorithm that performs the switching procedure discussed above on a leg basis, instead of mini chains. **Using the same example, we illustrate this ``leg-based switch''. Recall that we have two chains, 1-2-3-4-5-6, and a-b-c-d-e-f-g-h, with sub-chains 1-2-3, 4-5-6, a-b-c-d and e-f-g-h. Then, in ``mini chain-based'' switch, the pair (3,d) is the only potential switching point. However, assuming the compatibility of the legs the pairs (2,b) and (5,g) might be other potential switching points besides (3,d) as now we are allowed to break mini chains. Clearly, this leg-based switch is computationally more challenging compared to the previous case as the number of potential switch points are much higher and they may occur at some airport other than the BASE.** We carry out these iterations until we reach a local maximum point where we cannot improve the solution anymore. After reaching this local maximum, we perturb the solution by randomly breaking \alpha of the chains and creating new chains by patching these broken chains to other chains. Starting with this perturbed solution, we repeat the leg based local search algorithm. **The reason for implementing this perturbation is to avoid to be stuck at local optima**."

Note that we also carried out a sensitivity analysis on the value of the parameters such as n_iter, showing that the performance of our algorithm is robust with respect to different values of this parameter.

Minor Comments:

1) Page 2, line 25" "A particular cockpit crew is qualified to fly aircraft of only one fleet family": This is not really true. As pilots progress in their careers, they usually move to operating more advanced aircraft from less advanced, typically smaller to larger aircraft. When they move, they don't really lose the qualification for flying the smaller one but it is usually not a good idea to allocate them to the smaller one due to economic and other issues. So even though it is correct to say that each cockpit crew should be allocated to only one aircraft type, saying that they are not qualified for any others is not accurate. Please consider modifying accordingly.

We have changed the wording to reflect this more accurately. Now it reads "Typically a particular cockpit crew is assigned to fly aircraft of only one fleet family (due to qualifications and/or economic reasons), hence the crew scheduling problem can be decomposed into sub-problems for different fleet types."

2) Page 3, line 12: "In the literature, a daily schedule is assumed to repeat itself": I seriously doubt if this blanket statement is correct. There are many studies that assume this but also quite a few others that look at multi-day/weekly/monthly schedules. I urge the authors to check and adjust this language accordingly.

We reworded this sentence as: "The majority of the works in the literature focus on the North American flight network, assume that a daily schedule repeats itself with some minor changes, thus it is sufficient to solve a daily problem rather than the monthly problem and then adjust the solution to accommodate for these changes." Later in the text we also added and emphasized that there are weekly and monthly examples in the literature.

3) Page 3, line 16: "are considerably different.": Different from what? Please clarify.

We reworded this to read "flight patterns are considerably different than most of the works studied in the literature, mainly those related to the North American flight networks"

4) Page 3, line 22: "which is quite unlike North American networks": Again, while this is accurate for the large North American airlines, there are some others (e.g., Frontier Airlines, etc.), where I believe that this sentence does not apply. Please qualify these claims carefully.

We removed this part. We went over the manuscript and updated such blanket statements by adding qualifiers like "most airlines" etc. We thank the referee for pointing out these oversights in some of our statements.

5) Page 4, line 14: "For both problems, it is assumed that the flight schedule repeats daily": Again a big blanket statement that needs to be carefully qualified. There are a number of studies where this is not true. For example, see "Airline Crew Scheduling: Models, Algorithms, and Datasets" by Kasirzadeh, Saddoune and Soumis, 2014 (http://link.springer.com/article/10.1007%2Fs13676-015-0080-x) among others. Please be very careful making these claims without verifying if they are true.

Once again we thank the referee. We updated the sentence to reflect that this is the case in many works but not all.

6) Page 4, line 14: "In the case of exceptions, they are handled afterwards": Not clear what the authors mean by that. Please clarify.

This sentence now reads: "When extending the daily schedules to weekly and/or monthly schedules the exceptions to the repeating daily schedules are handled afterwards."

7) Page 5, line 2: "stations(cities)": Please consider using the word "airports" rather than "cities" because many airports have more than one cities associated with them and vice versa. In any case, "cities" is inaccurate since the airports are of concern in air transportation network optimization.

We updated the manuscript so that we refer to airports. Thanks for helping our exposition become more precise.

8) Page 5, line 40: "weekly assignments": This is contradictory to the previous claim that "In the literature, a daily schedule is assumed to repeat itself". Please correct the contradiction by fixing the previous issue.

Done, thanks.

9) Page 7, line 6: "airline problem research": Do you mean "airline scheduling research"?

We rephrased this to "airline schedule planning problems"

10) Page 8, first paragraph: This assumes that the passenger demand is inherently leg-based, which is an assumption that is not true. Spills, recaptures, itinerary based demand are some of the aspects that are ignored by the traditional leg-based fleet assignment and also by the authors here. But these have been incorporated by many studies so far (e.g., Wang, Klabjan, Shebalov, 2015; Barnhart, Kniker, Lohatepanont, 2002; Atasoy, Salani, Bierlaire, 2014). The authors need to at least acknowledge this shortcoming clearly here.

We thank the referee for pointing this out. We extended this section to emphasize that there are itinerary-based demand models in the literature that are better able to capture spills and recaptures.

11) Page 8, line 49: "and compatible": What does that mean? Please clarify.

We added (feasible connection time/location).

12) Page 9, line 9: Comma missing after L. Also, j,k \in L is redundant because of (j,k) \in L2. So I think you need to remove that.
Thanks for pointing this .We removed the redundant set L and added the comma.

13) Page 9, lines 14-35 provide a detailed math formulation but the rest of the paper does not seem to be using it directly anywhere. So I am not sure what exactly is the purpose of giving this formulation here. It appears that the later formulation on top of page 16 is related to this one but it is not clear what purpose is achieved by stating this one. Please clarify.

We provided the mathematical formulations in order to present a precise description of the problem. The model also enables the reader the better appreciate the complexity and the size of the problem at hand. The model (10)-(19) is the model for the integrated problem. (20)-(24) is the set partitioning problem formulation we used as part of our solution for the fleet planning problem. The IP in (25)-(31) is the model of the integrated problem (IFACPP) as solved in our algorithm. This particular model selects

and assigns chains and pairings so that each leg is covered by exactly one pairing and one chain while making sure that those two belong to the same aircraft family. Thus, the formulation (10)-(19) can be thought of as the generic formulation for the IFACPP, while (25)-(31) is the formulation used to solve the IFACPP with our solution approach. We have also updated the formulation (10)-(19) by adding two sets of constraints.

14) Page 9, constraints 15 and 16: They seem to be redundant to me. Isn't it true that these two and the w variables can simply be removed from the formulation without compromising either the optimality or the feasibility of the solution? In post-processing, you can always create w_pf variables for each y_p variable by setting one correct w to 1 for each y set to one. I can see these being useful in the LP relaxation though. Is that the purpose of having these? Do they provide valid inequalities? This is an important point needing clarification and explanation either way.

The reason for having w_pf variables and the related constraints in the formulation is as follows: We have aircrafts belonging two different aircraft families. Hence if a pairing is selected by this formulation (hence the corresponding variable y_p =1), then all the flights included in this particular pairing should be assigned to the same aircraft family (hence the corresponding x_aj variables). If it were not for these variables and the related constraints, the model may assign aircraft belonging two different families for the flights in the same pairing. It is not possible to decide this in the post-processing stage as this affects the objective function of the model as well.

15) Page 9, lines 14-35: The mathematical formulation is said to be for a month long schedule, but there is no attempt made to ensure the boundary conditions at the start and end of the month. Is it supposed to be a monthly repeating schedule? If so, then the aircraft balance at the beginning and end needs to be ensured by having appropriate constraints. If not, then what happens needs to be clarified. Later, on page 17 there is some mention of warm-up and cool-down period but I don't know what that means and how that is implemented. Please define everything carefully and justify your decisions in a mathematical rigorous way.

We assume that the problem is solved for each month separately, this is due to the non-repetitiveness of the schedules. In the model we made the following changes to reflect the "boundary conditions". We added s_a and t_a. s_a correspond to auxiliary flights that represents the start of the route flown by aircraft $a \in A$ (if any). The destination of this auxiliary flight is the origin airport of aircraft $a \in A$. t_a represents auxiliary flight that represents the end of the route flown by aircraft $a \in A$ (if any). This auxiliary flight is compatible with all flights. We added constraints (15) and (16). Constraints (15) require that there can be at most one departure from the auxiliary start flight for each aircraft. Similarly, constraints (16) require that an aircraft that has departed must end its route at the auxiliary end flight. These two conditions also ensure that there is a single path of flights for each aircraft, something that was lacking in our earlier formulation. We thank the referee for helping us notice this gap in our model.

The way we handled the boundary conditions in our computational study is explained next. To take into account the boundary conditions at the start of the month the initial position and condition of the aircraft and crew need to be known. Unfortunately we do not have this information. Indeed even

finding a feasible starting solution is a challenging task. The way we resolved this situation is to include a warm-up period. At day one we start assigning aircraft and crew to flights and progress from there on. At the end of the warm-up period we end up with a set of "initial conditions" and we solve the actual problem from thereon. In some sense we generate the boundary conditions with the warm-up period.

16) Page 11, line 6: What is L_hat? Also, what is the meaning of the notation L_hat=Intersection_SignL\l_k? Please clarify.

There was a typo, it should read L= L\l_k.

17) Page 11, line 16: "Concatenate feasible chains": How is that done? Please provide an algorithm for this.

The respective paragraph in the manuscript is modified as follows:

> We generate these chains in a greedy manner, that is, after adding the leg which arrives at time ta to the chain, we obtain the "ready time" as ta +r, where r is the minimum turn time, **which is 30 minutes.** Next, we identify the first fight leg that departs from that particular airport at or after the ready time and add that flight leg to the chain and we continue in this manner. **If there is no flight departing from that particular airport up until a certain time (14 hours for domestic and 48 hours for international locations), then there are no more feasible legs to be added to this chain** and we start a new chain from the first unassigned leg in the time sorted leg list. Once all legs are included to a chain, we concatenate feasible chains, to reduce the number of chains to the number of available aircraft. This concatenation step is different than adding legs to a chain in a greedy manner. **Here we include the possibility of repositioning of an aircraft and concatenate two previously generated chains if one aircraft can feasibly perform one after the other while accounting for the deadheading flight time from the arrival location of the last flight of the first chain to the departure location of the first flight of the second chain. As in the previous routine, concatenation process is also greedy hence a chain is concatenated to the earliest feasible chain. Note that the chains are not required to start and terminate at the same airport as the initial condition (locations of aircrafts, crews, etc.) at the beginning of each month might be different from one another.**

18) Page 11, last paragraph gives an example of subchains being created. It seems that the subchains are being made such that each subchain has to either start or end with the main chain. Is that always required as per the authors' definition of what constitutes a subchain? Please clarify and justify.

The subchains do not have to start or end with the main chain. Given two chain 1-2-3-4-5-6 and a-b-c-d-e-f-g-h-i-j-k-l it is possible to have an exchange like 1-2-3-d-e-f-g-h, and a-b-c-4-5-6-j-k-l, where h and 6 are also switching points.

19) Page 14, line 28: "pairings obtained in previous solutions": I think you mean to say "all previous solutions". If so, please add the vital word "all" which is really important to understand the implementation of this algorithm.

We added the word “all” in the text to clarify.

20) Page 14, line 31: What is "FP+". Please explain.

FP+ refers to the overall process to determine an FP solution. This process consists of four procedures, namely FP initialization, FP 0, FP chains update, FP LS, FP iterated local search, FP ILS, FP polishing, FP POL executed in a sequential manner. We also included a descriptive flowchart (Figure 1) to better present the solution approach.

21) Page 14, line 32: What do you mean by "traditional manner"?
Instead of “traditional manner” we now say “sequentially”.

22) Page 15, line 42: "The complexity of the problem is NP-Hard" is not an accurate statement. You can either say that the problem is NP-Hard or that the best known algorithm to solve this problem has an exponential complexity. Please revise.

Done, thanks. It now reads “the best known algorithm to solve this problem has an exponential complexity.”

23) Page 17, line 28: Why is the profit calculated by subtracting the cost of the least profitable aircraft? What is that supposed to do? It is highly unclear. Please explain and justify clearly.

We thank the reviewer for pointing this unclear sentence. What we meant to say was “In reporting these profits, we deduct the PROFIT of covering each leg with the least profitable aircraft, as this is a fixed PROFIT and does not depend on the solution”. We fixed this in the manuscript. The purpose of doing this is to better reflect the savings as the scale of profits/costs in 2 stages of the problem vary significantly.

24) In all the tables, and in sections 5 and 6 in general, please provide units of the costs, profits, and savings being reported. Are the dollars, euros, pounds etc.?

We thank the reviewer for bringing this omission to our attention. We added the units in all the tables.

Typographical Comments:
1) Page 2, line 9: Delete "of" in "Hence, simultaneously solving of all".
2) Page 10, line 10: Replace "with" in "results with an iterative solution" to "in".
3) Page 10, line 18: Delete "based" in "greedy based heuristic".
4) Page 13 constraint 21 has an obvious typo. Please fix it.
5) Page 15, line 33: Delete extra space after the word process on before the comma.
6) Page 15, line 38: Replace "select" with "selects".

7) Page 15, line 48: "to over by': Please fix the typo. Do you perhaps mean "to be covered by"?
8) Page 16, line 25: Replace "guarantee" with "guarantees".
9) Page 17, line 16: Replace "Ghz" with "GHz".
10) Page 17, line 31: Replace "Finally, The" with "Finally, the".
11) Page 18, line 39: Delete the word "thousand".

We thank the referee for pointing out these typos, we fixed them.

Reviewer #2: As the authors note, airline planning problems are computationally challenging and decomposition, while improving tractability, can negatively impact solution quality. I am therefore quite interested in their research topic. In its current form, I do not find the paper to be appropriate for publication. I encourage the authors, however, to revise and resubmit; I would look forward to the opportunity to review a revised manuscript. My primary concerns are as follows:

1) It was not clear to me what motivated this research. Is this a real or academic problem? Is it common in the European sector of the industry to have non-repeating schedules? What types of carriers would benefit from this research?

The particular variant of the Fleet Assignment and Crew Pairing Problem we study in this paper arises in many airlines, in particular in airlines where crews are paid based on a fixed salary. The aperiodicity of the flight schedule, a feature common in many European airlines, requires addressing the monthly problem, which makes the problem size significantly big. We show that by taking into account the network structure, in this case a single base with many point to point flights, we are able to generate good solutions in reasonable time. The particular EAC problem allows us to demonstrate our approach in a real-world setting. We added comments in the introduction and conclusion section to emphasize the generalizability of our approach. We thank the reviewer for allowing us to better highlight our contributions

2) Does the problem need to be solved at a monthly level? Why was this time horizon chosen -- because of crews? Given that there is only one hub to which flights often return, could the problem be decomposed to a shorter time period with limited degradation on the solution quality?

There are many advantages of considering the monthly problem over considering the weekly problem. First, in practice, the fleet assignment is scarcely periodic. Indeed, the daily demand on some legs may experience large variations due to some special events. To match these demand fluctuations, fleet planner may assign to the same flight leg different aircraft types at different weeks or days of the week. In addition, aircraft should undergo various types of periodic maintenance and therefore the set of aircraft that are available at a given week is varying.

3) Why is it necessary to model individual aircraft rather than fleet (sub) types? Presumably this would lead to symmetry issues negatively impacting convergence. And why is the problem modeled as it is, rather than with a more traditional time-space network formulation? Presumably this is what makes the problem so big and necessitates the heuristic solution approach. The reason for the model choice was not clear to me.

A first advantage is that the aircraft routes are built together with the solution. Therefore, the model guarantees the generation of feasible aircraft routes. Second, some aircraft might require undergoing thorough inspections. This type of planned maintenance requires the aircraft to remain grounded for a

long period of time (see response to Referee 3). In this case, the aircraft will not be available for flowing some legs. A minor modification of (C1) can easily accommodate this type of restriction.

4) Given the number of flights that go into a station and then there is only flight out, it is not clear where the opportunities for major improvement are, ie it would seem that crew typically follow the aircraft. Can you better argue for the potential improvement in costs by solving simultaneously rather than sequentially? How big is this impact?

Our computational results (Table 3) show that for the five instances we have solved, the simultaneous solution method improves the cost contribution of the Crew Pairing Problem by $24,545 to $827,209, with an average of $263,271. In addition, the profit contribution of the Fleet Assignment component improves by $2,206,347 to $2,604,832, with an average of 2,360,407. This results in an improvement of $2,255,304 to $3,432,041, with an average of $2,623,678 per instance, a significant amount by any measure. In addition, low cost crew schedules also correspond to higher quality crew schedules, since crew members do not like to perform deadheadings or layovers. Improved employee satisfaction leads to improved customer satisfaction, and ensures that the airline does not lose any customers due to low quality of service.

5) If I understand correctly, you did not have access to real-world cost data. Can you please elaborate on how you generated your cost data?

We used representative industry data for aircraft operating costs. These are available online from various resources. We used an online Aircraft Cost Evaluator tool (https://www.conklindd.com/p-33-aircraft-cost-evaluator.aspx). Although these costs may not be exact costs incurred by EAC they would be representative in terms of order of magnitude and relative proportions with each other. We added a brief explanation in the manuscript about this.

6) Presumably, the sheer size of your formulation is what motivates you to solve the problem heuristically. Section 4 is quite dense -- many pages of algorithms. I would have benefited from a much higher level overview of the approach at the beginning to put all of these details into context.

We thank the reviewer for helping provide a better presentation of the algorithms. We included a descriptive flowchart (Figure 1) to better present the solution approach.

7) I also would appreciate a much clearer and more concise argument about the quality of the heuristic and justification for it over an optimal approach.

The standard approaches of using a multi-commodity network flow problem formulation for the Fleet Assignment component, and a Set Partitioning Problem formulation for the Crew Pairing Problem component fail for the sheer size of the problem we aim to solve. The integrated problem is clearly out of the reach of an exact algorithm, for the state-of-the-art hardware and software.

We have implemented an upper bound using a multi-commodity network flow problem formulation for the Fleet Assignment component, and ignored the cost of the Crew Pairing Problem component, the details of which are provided in the paper. Based on the results of this upper bound, we now show that

our solution to the IFACPP is on average are 8% away from the upper bound of the Fleet assignment problem. We now present these results in the Computational Results section.

I hope to read a revised version of the manuscript in the future that enables me to more clearly appreciate the contents and contributions of this interesting research.

Reviewer #3: Summary
In this paper, the authors consider an integrated monthly scheduling problem consisting of the fleet assignment and crew pairing problems of the traditional airline scheduling process. The solution approach is highly related to a specific case of a European airline with non-periodic schedules in a hub-and-spoke network. The solution approach combines fleet assignment and crew pairing by using a set of heuristics and exact mathematical optimization approaches. Therefore, an overview of studies tackling either monthly or integrated resource scheduling problems is provided. Afterwards, the problem definition and the solution approach is presented and evaluated by computational experiments for five instances. The results show that the problem can be solved in a reasonable period of time and the integrated approach leads to better results than the traditional sequential approach.

General recommendation:
The paper addresses a real world problem with a strong practical connection. According to the authors, the main contribution is the technical solution approach, integrating two airline planning stages. The problem characteristics (especially the network structure) of the given monthly scheduling problem is addressed as one of the main motivations for the approach. However, these problem characteristics could be taken into consideration in more detail in the modeling step as well in the discussion of the computational results.

The results are presented in a rather short way and do not always coincide with the questions emerging in the introduction. As an example, crew costs are referred to as the second largest of an airline. However, the crew salaries /paid time are not taken into account in the results. Also the sub-optimality of scheduling stages due to the traditional sequential solution approach can be addressed explicitly in the results. (It does not necessarily mean that e.g. the crew salaries should be taken into account but the authors should refer to the problem definition in more detail.)

I suggest a major revision mainly considering following aspects:

Major comments

p.3&19 You mention that (actually neglected) aircraft maintenance restrictions are satisfied although not explicitly considered. It is not clear to me if it applies for all maintenance restrictions? Can you include some figures in order to provide more details? I think it is an important fact that takes into account implications of the underlying specific network structure.

Airlines utilize a planned maintenance program that includes checks having different levels of detail. These checks are often referred to as A-checks, B-checks, C-checks, and D-checks. The most frequent, and least comprehensive, of these checks are the A-checks. On the other hand, D-checks are extremely comprehensive, involving thorough inspection of systems and components. They might occur only three to six times during the lifetime of an aircraft.

The periodicity of these checks depends on the aircraft type and airlines' internal rules. For instance, at Lufthansa the A-check are carried out every 350 to 750 flying hours (see http://www.lufthansa-technik.com/maintenance). In addition to general inspections of the interior and the aircraft hull, it also covers service checks as well as engine and function checks. At the same time the technicians replenish oil, water and air and eliminate defects whose rectification has been postponed on the grounds that they did not impair flight safety. Depending on the requirement, the A-check will take between 45 and about 260 man-hours. Since A-checks are usually scheduled overnight, when the aircraft is idle, they are often ignored in integrated airline tactical models. It is noteworthy that the proposed model can easily accommodate the situation where an aircraft needs be grounded for a longer time to undergo a thorough preventive maintenance.

p.3 The solution approach tackles a scheduling problem that is clearly specialized on a particular form of an airline's flight schedule. In how far the approach is - as you mention - tailored to this one problem or can be generalized to other airline scheduling settings?

The particular variant of the Fleet assignment and Crew Pairing problem we study in this paper arises in many airlines, in particular in airlines where crews are paid based on a fixed salary. The aperiodicity of the flight schedule, a feature common in many European airlines, requires addressing the monthly problem, which makes the problem size significantly big. We show that by taking into account the network structure, in this case a single base with many point to point flights, we are able to generate good solutions in reasonable time. The particular EAC problem allows us to demonstrate our approach in a real-world setting. We added comments in the introduction and conclusion section to emphasize the generalizability of our approach. We thank the reviewer for helping us better present our contributions.

p.3 Closely related to the previous comment - are there findings in retrieving the special problem characteristics that can be generalized in terms of special modeling/solution techniques? Or, more generally, please state clear the theretical contribution of this work.

We demonstrate that the traditional approach of solving the Fleet Assignment and Crew Pairing Problems can be severely suboptimal, and iteratively solving these two problems with an information feedback mechanism between the two components can improve the net profit up to $1,000,000, albeit with erratic results. We also demonstrate that it is possible to simultaneously solve these two components, which consistently improves the net profit by $2,255,304 to $3,432,041, with an average of $2,623,678 for all instances, and requires a much shorter CPU time.

p.7 The literature review is kept very short, some additional references could be included though. However, an analysis of the existing approaches and a subsequent discussion of the research needs the new approach is tackling, is missing. It is mentioned at the end of the Introduction as well as in the beginning of chapter 2. Nevertheless, a final conlusion after the literature review is essential. Which techniques can be reused? What are the actual differences to the new approach?

We thank the reviewer. We expanded the literature review section to better summarize existing works in the literature. We added a paragraph at the end of the literature section about strength and weaknesses of the available methods for the given IFACPP studied in this paper.

p.8 You say that '[The higher number of flight legs] makes it impossible to use the traditional daily models.' For sure, the number of flights of an instance is a difference between daily and monthly schedules. However, can you expose expose the actual differences between the literature approaches and the proposed approach from a more technical point of view?
Why exactly the daily-based scheduling approaches cannot be used for the monthly problem? Can you add a discussion on possibilities (or impossibilities) concerning the usage of approaches from the literature to cope with the monthly problem, e.g. by splitting up the schedule in smaller sub-schedules.

There are many advantages of considering the monthly problem over considering the daily/weekly problem. First, in practice, the fleet assignment is scarcely periodic. Indeed, the daily demand on some legs may experience large variations due to some special events. To match these demand fluctuations, fleet planner may assign to the same flight leg different aircraft types at different weeks or days of the week. In addition, aircraft should undergo various types of periodic maintenance and therefore the set of aircraft that are available at a given week is varying.

p. 9 You say that the model contains over 94 billions of binary variables. Of course, this problem complexity arises when integrated problems are addresses. Complexity can be tackled by both modeling and the solution approach. In the previous sections you explicitly mention the special characteristics of the network structure which is apparently one of the most significant motivations for your approach. Besides some main characteristics of the flight network, there are no further details mentioned that would possibly lead to a better insight into the problem. Instead, the problem is modeled straightforward without taking into account the special network structure. But aren't there network characteristics that may lead to an a priori reduction of the model complexity here? Or at least you can explain why this is not the case. A more detailed case study may improve the papers' quality at this point.

Here are some specific network characteristics that rendered the problem more tractable and made it feasible to solve the large-scale integrated model:

- **We used the rotation concept (sequence of consecutive flights that are always flown by the same crew/aircraft to significantly reduce the network size,**
- **The objective function of the CPP is much easier than the pay-and-credit cost that is widespread in North American (and in the literature as well),**
- **Since the very great majority of preventive maintenance checks are performed overnight when aircraft are idle, the aircraft maintenance routing constraints play a mild role and were therefore ignored at the tactical level.**

pp.17 Can you describe the five instances more detailed? Why did you choose these instances? Are there differences in the network structure, e.g. due to difference seasonal components (summer/winter)? This may be important as the instances imply a high deviation in terms of benefits from the novel approach (the quite significant average improvement values mainly depends on particular instances).

How your methods would perform with more than two sub-fleets?

We demonstrated our approach on several instances. This is due to the fact that schedules are not repetitive for the EAC. Based on the month/season studied the total number of legs change as well as the distribution of flights within the flight network. Overall there is a 5% change in the total number of flights studied in the instances, this accounts for more than 1000 legs difference. Furthermore, based on the month, we see that some new destinations are added and some others see significant changes in the number of flights schedules to and from those airports. The increase in the number of flights to an airport can be as high as 71% as the season changes. The average increase in the number of flights to an airport with changed number of total flights is 31%. Furthermore, using several instances allowed us to test our approach in the face of changing schedules.

We studied 2 fleet types, but accommodated for 5 subtypes within one fleet and 8 subtypes within the other fleet type. In the EAC two fleet families are used for short-medium haul flights. Long haul flights and their fleet were not included in the study as their scheduling typically required round-trip flights with minimal or no connections. In the case of more than two fleet families, the computation of "benefit" in the FP stage would include the other alternate families as well. We do these both at chain level and leg level, having more than 2 fleets would not change the structure of the algorithm at all. In the CPP stage, instead of solving 2 sub-problems we would need to solve *n* sub-problems, where *n* is the number of fleet families. Again the structure of the algorithm would not change. As expected run times may decrease as we will be solving n sub-problems, however the sizes of the sub-problems will be relatively smaller, which may be considered as a decomposition of a larger problem and may improve solution times

pp.14-15 Do we know whether (and why) the iterative process converges? Please provide a more careful analysis of convergence behaviour.

As can be seen in the computational study section of the paper we carried extensive sensitivity analysis on the procedures. These results are presented in Figure 2, where we clearly see that our algorithms converge after a certain number of iterations.

Minor comments

p. 2 Airline planning consists of more than flight schedule design, this is rather the first step of the tactical planning. You can either mention the flight schedule design as the first step in tactical planning or point out more details on the network and frequency planning (which is more a strategic task).

We updated the wording so that it now reads "airline schedule planning problems". We hope this distinguishes the problems we are considering from the more general strategic ones.

p. 3 The 'rotation flight type' can be simply described by the out-and-back principle, see e.g. Kohl et al. (2007). Is there a significant difference to that definition?

"Rotation flight type" is similar to "out and back" principle mentioned in Kohl et al. (2007). There is not a very significant definition. When we say rotation, we refer to a sequence of flights that are always operated by the same crew and aircraft. Out and back type flights are an example of this. The rotation definition is aimed to cover other constant sequences as well. In this paper, however, we focused mainly on "out and back" type of rotations.

p. 5 rephrase: '…can be deleted to create islands, see…'

Done, thanks.

p. 8	Where do the expected profit values come from? Is there historical data or do you use forecasts of the airline?

For the revenue, we tried to obtain representative figures by sampling from prices offered on the EAC's website for different day, times and flights. We used a simple, leg based model that capture the effects of time of day, domestic vs. international and average demand (approximated by the number of flights scheduled). As we are trying to provide a proof of concept we focused on getting representative revenue data rather than the exact revenue. Similarly for the costs we used representative industry data for aircraft operating costs. These are available online from various resources. We used an online Aircraft Cost Evaluator tool (https://www.conklindd.com/p-33-aircraft-cost-evaluator.aspx). Although these costs may not be exact costs incurred by EAC they would be representative in terms of order of magnitude and relative proportions with each other. We added a brief explanation in the manuscript about this.

p. 8	The fact that 80% of all flights depart/arrive at the hub as well as the abbreviation 'BASE' is already mentioned in the Introduction. The same holds for 'rotation flights'. In general, many descriptions in the beginning of chapter 3 are repetitions of chapter 1.

As we are describing the specific problem, for sake of completeness we wanted to include all problem specific details together. This way we also remind the reader about our terminology. However we shortened some of the explanations in the introduction to minimize repetition.

pp. 8	In the previous models (1)-(6) and (7)-(9) you first present the model and afterwards the parameters. In terms of consistency, you should do that in model (10)-(17), too.

Done, thanks.

p.17	It is not clear what you mean with 'representative figures'. Can you describe what exactly are the 'proportions and magnitudes' here? Is there no real data available and how the estimation of the representative figures is performed?

For the revenue, we tried to obtain representative figures by sampling from prices offered on the EAC's website for different day, times and flights. We used a simple, leg based model that capture the effects of time of day, domestic vs. international and average demand (approximated by the number of flights scheduled). As we are trying to provide a proof of concept we focused on getting representative revenue data rather than the exact revenue. Similarly for the costs we used representative industry data for aircraft operating costs. These are available online from various resources. We used an online Aircraft Cost Evaluator tool (https://www.conklindd.com/p-33-aircraft-cost-evaluator.aspx). Although these costs may not be exact costs incurred by EAC they would be representative in terms of order of magnitude and relative proportions with each other. We added a brief explanation in the manuscript about this.

Annals of Operations Research manuscript No.
(will be inserted by the editor)

Solving a Large-Scale Integrated Fleet Assignment and Crew Pairing Problem

Okan Örsan Özener · Melda Örmeci Matoğlu · Güneş Erdoğan · Mohamed Haouari · Hasan Sözer

Received: date / Accepted: date

Abstract Airline **schedule planning** problems are typically decomposed into smaller problems, which are solved in a sequential manner, due to the complexity of the overall problems. This results in suboptimal solutions as well as feasibility issues in the consecutive phases. In this study, we address the Integrated Fleet Assignment and Crew Pairing Problem (IFACPP) of a European Airline. The specific network and cost structures allow us to develop novel approaches to this integrated problem. We propose an optimization-driven algorithm that can efficiently handle large scale instances of the IFACPP. We perform a computational study on real-world monthly flight schedules to test the performance of our solution method. Based on the results on instances with up to 27,500 flight legs, we show that our algorithm provides solutions with significant cost savings over the sequential approach.

Keywords Fleet assignment · Crew pairing · Airline operations planning · Large-scale neighborhood search

This project is funded in part by TUBITAK grant 110M308.

Okan Örsan Özener
Ozyegin University, School of Engineering, Istanbul, Turkey
E-mail: orsan.ozener@ozyegin.edu.tr

Melda Örmeci Matoğlu
University of New Hampshire, Peter T Paul College of Business and Economics, Durham, NH, USA
E-mail: Melda.OrmeciMatoglu@unh.edu

Güneş Erdoğan
University of Bath, School of Management, BA1 7AY, United Kingdom
E-mail: g.erdogan@bath.ac.uk

Mohamed Haouari
Department of Mechanical and Industrial Engineering College of Engineering, Qatar University, Doha, State of Qatar
E-mail: mohamed.haouari@qu.edu.qa

Hasan Sözer
Ozyegin University, School of Engineering, Istanbul, Turkey
E-mail: hasan.sozer@ozyegin.edu.tr

1 Introduction

Airline **schedule planning** problems have attracted significant attention from the operations research community over the past years, due to their sheer size and inherent complexity. These problems not only provide interesting challenges from a theoretical standpoint, but also offer opportunities for significant economical benefits for airlines. As the scope of these planning problems widens, the potential benefits from effectively solving them increase as well. Hence, simultaneously solving all related planning problems can potentially result in the maximum benefit for the airlines. However, past a certain point, the resulting integrated problem becomes so complex that it is no longer efficiently solvable. In airline planning context, this happens quite often as each individual problem is already hard to solve due its magnitude (thousands of flight legs, crews, etc). Therefore, the traditional approach is to sequentially solve these planning problems.

The first stage in airline **schedule planning** is the construction of the monthly flight schedule. The flight schedule consists of a list of flights to be operated in a given month. This schedule usually depends on the expected demand for the flight segments as well as the fleet size of the airline. Next, the *Fleet Assignment Problem* (FP) is solved, where aircraft types are assigned to the scheduled flight legs based on their capacities and operating costs, with the objective of maximizing the net profit. In the next step, maintenance requirements of the aircraft are considered while individual aircraft are assigned to specific routes. This is called the *Aircraft Routing Problem*.

Next, the airline must assign cockpit and cabin crew for each of the scheduled flights, which is referred to as the *Crew Pairing Problem* (CPP). **Typically a particular cockpit crew is assigned to fly aircraft of only one fleet family, due to qualifications or economic reasons**, hence the crew scheduling problem can be decomposed into sub-problems for different fleet types. The objective of the crew scheduling problems is to minimize the crew related costs. The feasibility of a crew schedule depends on rules imposed by government, international organizations (e.g. International Civil Aviation Organization), labor unions, and airline companies themselves. For a global airline that operates tens of thousands of flights in a month and employs thousands of cockpit and cabin crews, even determining a feasible crew schedule is a daunting task. The process consists of two stages. First, in the crew pairing stage, a set of feasible and efficient pairings is generated to cover each flight leg. Next, in the crew assignment stage, these pairings are used for creating monthly rosters to be flown by the available crews.

This sequential approach clearly leads to suboptimal solutions as well as potential feasibility issues. For example, after the flight schedule is created, airlines determine a fleet assignment by assigning an aircraft type to each leg, while considering the aircraft availability at each decision point. Then, these flights are assigned to individual aircraft while making sure that there is a pathwise feasible routing that also satisfies maintenance constraints of the aircraft. At the next stage, the output of the routing problem is used as an input to the crew scheduling problem. As the constraints and cost factors of the crew scheduling problem have not been considered in the previous stage, it is quite likely for airlines to face infeasibility issues or end up with less efficient solutions.

For commercial airlines, fuel costs constitute the largest chunk of the total operating cost, whereas the crew costs are a close second (Gopalakrishnan and

Johnson (2005)). The fleet assignment decision plays an important role in total fuel costs incurred by an airline just like the crew planning decision affects crew related costs. As the former decision heavily influences the latter decision, we attempt to tackle both problems simultaneously to determine a better solution for the overall planning problems of an airline. Even though individual FPs and CPPs have been studied extensively in the literature, integrated approaches are quite limited; these are summarized in detail in Section 2.

In this paper we address the Integrated Fleet Assignment and Crew Pairing Problem (IFACPP) of a European Airline Company (EAC). The associated FP **and CPP** for this particular case differs from the ones studied in the literature **in three important aspects: (i) schedule periodicity, (ii) flight network structure, (iii) cost structure. These features are not limited to EAC but arise in airlines worldwide. Firstly, the majority of the works in the literature focus on the North American flight network, assume that a daily schedule repeats itself with some minor changes. Thus, it is sufficient to solve a daily problem rather than the monthly problem and then adjust the solution to accommodate for these changes.** Due to the irregularities in the flight schedule, employing these techniques proposed in the literature for may not produce efficient solutions for **most European airlines and** the problem under consideration.

Secondly, the flight network has its own characteristics as well. There is only one crew base (we refer to it as BASE) and more than 80% of the flights either originate from or terminate at BASE. In addition to this, a significant portion of the flights to several domestic and most international destinations are *rotation* type flights. For example, if there is a flight from BASE to Rome in the morning on a given day, there is also a flight returning to BASE later that day, as there exists no other flights from Rome to any other destination and the aircraft (hence the crew) will not stay in Rome for an additional day.

Finally, the cost structure as well as the number of hubs and flight patterns are considerably different **than most of the works studied in the literature, mainly those related to the North American flight networks. In EAC,** like most European airlines, the crews receive fixed monthly salaries, thus the objective of the CPP stage is to minimize the costs of *layovers* (crews resting overnight in locations other than their bases) and *deadheadings* (crews being transported on a flight as passengers). **In most North American airlines the crew cost structure is based on complex rules that depend on flying hours, time away from base, etc.** Based on these characteristics of the problem, we attempt to develop a specific solution approach to the IFACPP under consideration.

Our objective is to solve the *monthly* IFACPP, which involves approximately 27,000 flights, two aircraft families, over 100 aircraft and airports. We do this, in part by relying on the CPP solution approach developed by Erdogan et al. (2015). We develop a heuristic approach to the FP. Our model completely integrates the FP and CPP, but neglects aircraft maintenance constraints of aircraft routing component. As in Sandhu and Klabjan (2007), in the case where aircraft maintenance requirements are satisfied due to the structure of the underlying flight network, our model also integrates the routing problem. In our implementation, we employ a combination of heuristic and exact methods, including greedy heuristics, local search methods, matheuristics, assignment models, set covering models, and finally an integer programming model. The heuristics allow computational efficiency for a problem of this complexity and size, and exact methods working in combination with the heuristics improve the quality of the solution. Our major contribution is the development of a computationally efficient solution to the monthly integrated problem. Although there are a small number of studies on "monthly" or "integrated" problems, to the best of our knowledge our study is the first to tackle both aspects simultaneously. **We show that, by taking into account the special structure of the problem we are able to generate good solutions to such a large sized problem, and that by addressing the integrated problem better results are obtained than addressing the individual problems sequentially. Our approach is not limited to the airline schedule planning problem at EAC, it can be applied to other airlines with similar cost and network structure.**

The outline of the paper is as follows. In Section 2, we summarize the existing studies in current literature. In Section 3, we provide the problem definition. In Section 4, we describe the solution approach, and in Section 5, we present the results of our computational study for 5 instances each with approximately 27,000 legs. Finally, in Section 6, we provide our conclusions.

2 Literature Review

Both FP and CPP have been studied widely in the literature. For both problems, it is **a common assumption of the literature** that the flight schedule repeats daily, thus most models are developed to tackle the daily problem. **When extending the daily schedules to weekly or monthly schedules, the exceptions to the repeating daily schedules are handled afterwards.**

In FP, given a flight schedule and a set of aircraft, the objective is to assign each flight to an aircraft type so that the profit is maximized. These aircraft types may have different capacities, availabilities and operating costs. Assigning an aircraft with a capacity higher than demand will result in unsold seats, while assigning an aircraft with a capacity smaller than needed on a flight will result in lost customers. The operating costs (including fuel, landing fees etc.) also depend on the type of aircraft assigned to the flight. FP has been largely treated as a multi-commodity network flow problem with cover, flow balance, and aircraft availability constraints. We refer the reader to Abara (1989); Daskin and Panayotopoulos (1989); Berge and Hopperstad (1993); Subramanian et al. (1994). Typically, the daily schedule is represented as a closed-loop space-time network, where a directed arc corresponds to the movement of an aircraft on a flight leg, and each arrival and departure at an **airport** is depicted by a node for each valid aircraft type. Aircraft waiting on ground is depicted by an arc between successive nodes in time. Overnight stays are depicted by wrap-around arcs from the last node to the first node at each **airport**. **For a survey of FP models see Gopalakrishnan and Johnson (2005).**

The traditional fleet assignment model (an adaptation of Hane et al. (1995)) is stated below.

$$\max \sum_{f\in F}\sum_{\ell\in L} p_{f\ell}x_{f\ell} \tag{1}$$

$$\text{s.t.} \sum_{f\in F} x_{f\ell} = 1 \qquad \forall \ell \in L \tag{2}$$

$$\sum_{o\in S} x_{fost} + y_{fst^-t} - \sum_{d\in S} x_{fsdt} - y_{fstt^+} = 0 \qquad \forall fst \in N \tag{3}$$

$$\sum_{\ell\in O(f)} x_{f\ell} + \sum_{s\in S} y_{fst_nt_1} \leq A_f \qquad \forall f \in F \tag{4}$$

$$x \in \{0,1\} \tag{5}$$

$$y \geq 0 \tag{6}$$

where the parameters and decision variables are defined as follows:

S : the set of **airports** in the network

F : the set of aircraft (fleet) types%, indexed by f,

L : the set of flight legs scheduled, indexed by ℓ or odt, where $o, d \in S$, and t is the take off time at o or the ready time for the next take-off at d

N : the set of nodes in the network, indexed by fst, where $f \in F$, $s \in S$, and t denotes the take off or landing time at s,

$O(f)$: the set of arcs for fleet type $f \in F$ that cross the aircraft count time-line,

$p_{f\ell}$: the profit of assigning fleet type f to leg ℓ, $f \in F, \ell \in L$,

A_f : the number of available aircraft of each aircraft type f, $f \in F$,

t^- : the time preceding t in the event time-line.

t^+ : the time following t in the event time-line.

t_n : the time of the last node in the time-line before the aircraft count time-line

t_1 : the time of the successor to the last node in the time-line before the aircraft count time-line

The decision variables are x and y, where,
$x_{f\ell} = 1$, if aircraft type f covers leg ℓ; $f \in F, \ell \in L$; and 0, otherwise.
$y_{fstt'}$ is the flow of aircraft on the ground arc from node $fst \in N$ to node $fst' \in N$ at **airport** $s \in S$ in aircraft type f's network, for $f \in F$, where $t' = t_1 \leq t = t_n$ for wrap-around arcs, and $t' > t$ otherwise.

The objective function (1) maximizes the profit of assigning aircraft types to flight legs. Constraints (2) ensure that each flight in the schedule is assigned exactly one aircraft type, constraints (3) conserve the flow of aircraft (an aircraft that takes off must land), and constraints (4) ensure that the total number of aircraft assigned cannot exceed the number available in the fleet.

For a network of hundreds of **airports** and thousands of legs, this problem becomes very challenging to solve. Indeed, Gu et al. (1994) have shown this problem to be NP-hard for three aircraft types, even without the availability constraints. To this end, there have been efforts at reducing the problem size by using preprocessing techniques such as *node aggregation* and creation of *islands*. In node aggregation, irrespective of the exact time of arrival or departure, consecutive arrivals and the subsequent consecutive departures can share a single node such that each arrival at the aggregated node can be feasibly connected to any departure at this node, see e.g. Hane et al. (1995). In the case of sparse flight activities where no aircraft are on the ground during certain periods of time, these ground arcs with zero value can be deleted to create islands, **e.g. see Rushmeier and Kontogiorgis (1997)**.

Aside from the basic fleet assignment model, several variations and extensions (such as integrating aircraft maintenance, discretized time windows, weekly assignments with homogeneity etc.) have been studied in the literature. The methods used to tackle these problems include solving the linear programming (LP) relaxation of the mixed integer model and applying rounding heuristics and using branch-and bound search techniques for remaining variables (Rushmeier and Kontogiorgis (1997)), Lagrangean relaxation (Daskin and Panayotopoulos (1989)), branch-and-price solution schemes (Hane et al. (1995); Bélanger et al. (2006)), large-scale neighborhood search (Ahuja et al. (2007)). We refer the reader to Sherali et al. (2006) for a more comprehensive survey of fleet assignment models.

CPP is generally considered as the hardest of the airline planning problems and has been extensively studied in the literature. Usually, the CPP is modeled as a set partitioning problem where the rows represent flights to be covered and the columns represent the candidate crew pairings. The objective is to minimize the total cost of pairings while covering each flight.

We now provide a basic mathematical representation of the CPP.

$$\min \sum_{j \in P} c_j x_j \tag{7}$$

$$\text{s.t.} \sum_{j \in P} a_{\ell j} x_j = 1, \forall \ell \in L \tag{8}$$

$$x_j \in \{0, 1\}, \tag{9}$$

where the parameters and decision variables are defined as follows:
P: the set of all feasible pairings,
c_j: the cost of pairing j,
$a_{\ell j}$: 1, if pairing j covers flight ℓ, and 0 otherwise,
L : the set of all flights that must be covered in the period of time under consideration.
The decision variable x_j, takes the value of 1 if pairing j is selected, and 0 otherwise.

Numerous solution methodologies have been developed for CPP. Many of these approaches rely on pairing generation and pairing selection strategies. **In early works heuristic approaches were employed to generate a subset of the pairings (e.g.** Arabeyre et al. (1969); Gershkoff (1989); Anbil et al. (1991)) **that were then used in the set partitioning problem. The downside of this approach is that it is not possible to quantitatively assess the quality of the solution.** Pairings are frequently generated using either with an enumerative algorithm or shortest path approximation approach on a graph network that can be represented either as a flight segment network or a duty network (Paleologo and Snowdon, 2007). **In these, branch-and-price techniques have been used frequently (Ryan; Barnhart et al., 1994; Vance et al., 1997; Barnhart et al.,**

1998b; Desaulniers, 1998; Butchers et al., 2001; Dück et al., 2011). However, pricing algorithm may take prohibitively long time for very large instances.

Many works in the literature address the daily CPP problem with instance sizes of up to 1000-2000 daily flights, there are others that address the weekly (e.g. Butchers et al. (2001) for instances from 300 to 900 legs), monthly (e.g. Kasirzadeh et al. (2015) for instances from 1013 to 7765 legs, and Subramanian and Sherali (2008) for instances from 2000 to 16000 legs) problem. Large-sized monthly crew pairing instances are usually addressed by a three-phase approach in the industry. Initially, assuming the daily schedule repeats, a daily problem is solved. Then in the second phase the a weekly solution is derived from the daily solution and finally exceptions in the monthly schedule are handled starting from an initial solution obtained in the weekly problem. This approach enables handling large instances in shorter computational times and favors regularity in the solution. However, when flight schedules are not regular this approach yields poor solutions. Indeed, Saddoune et al. (2013) develop a solution method on a rolling horizon basis that performs better than this multi-phase approach.

Another approach used to solve the CPP is heuristics. Metaheuristics like simulated annealing (Emden-Weinert and Proksch, 1999), tabu search (Cavique et al., 1999), genetic algorithms (Chen et al., 2013; Levine, 1996; Ozdemir and Mohan, 2001), ant colony algorithms (Deng and Lin, 2011), particle swarm optimization (Azadeh et al., 2012, 2013) have been employed, as well as hybrid algorithms that combine that combine heuristics with exact optimization methods (e.g. Panayiotis et al. (2000); Aydemir-Karadag et al. (2013); Erdogan et al. (2015)). Advantages of heuristics include speed and the size of instances that can be handled, whereas the quality of the solutions obtained can only be empirically demonstrated. We refer the reader to Desaulniers (1998); Barnhart et al. (2003a,b); Gopalakrishnan and Johnson (2005); Kasirzadeh et al. (2015) and references therein **for a more comprehensive review.**

Recently there have been attempts that aim to integrate several of the airline problems. There have been efforts in jointly solving the flight scheduling problem and FP. For example, Rexing et al. (2000) develop a version of the basic FP model that assigns a time window to each flight, which makes it possible to optimize departure times. Lohatepanont and Barnhart (2004) start from a base schedule and introduce modifications to the base schedule. Different demand parameters and demand correction terms are estimated and revised iteratively. In addition to a modified schedule, they determine an associated fleet assignment. Sherali et al. (2009) develop a model that integrates flight scheduling and fleet assignment using itinerary-based demands, for which they apply Benders decomposition to a mixed integer programming model augmented with several classes of valid inequalities and **Sherali et al. (2013) address a similar problem that also includes flexible flight times, optional legs, and multiple fare-classes.**

Clarke et al. (1996) address a variant of the FP with maintenance and crew considerations. They include constraints and modeling aspects in the basic fleet assignment model that retain its solvability and improve solution quality. Desaulniers et al. (1997) consider a daily aircraft routing and scheduling problem, for which a heterogeneous fleet has to be assigned to a set of legs with departure time windows. They present two formulations, one based on set partitioning and the other on a multi-commodity network flow model. Exploiting the equivalence between the two formulations, they propose branching strategies. Barnhart et al. (1998a) address the integrated fleeting and routing problem for which they present a string-based model and branch-and-price algorithm, where strings refer to a set of connecting flights starting and ending at maintenance stations. More recently, Haouari et al. (2011) propose two exact algorithms to the integrated fleeting and routing problems. Their algorithms are based on a tailored Benders decomposition algorithm that includes Benders cuts, maximal clique constraints and combinatorial cuts that help eliminate symmetry; and column generation (i.e. branch and price) algorithm with branching on the path variables.

In another track of integrated airline **schedule planning problems**, the integrated routing and CPP has been studied. For example, Cordeau et al. (2001) have utilized linking constraints to ensure that for short connection times, crew does not change aircraft. The authors solve the problem using Benders decomposition, where the algorithm iterates between a master routing problem and a crew pairing sub-problem. Klabjan et al. (2002) have partially integrated routing and crew pairing problems. They solve the crew scheduling problem before the aircraft routing and assume that flight departure times are not fixed, but must be performed within a time window. To ensure feasibility of the routing problem, they add additional constraints to the crew scheduling model. Cohn and Barnhart (2003) solve an integrated routing and crew pairing problem where they include variables for complete routing solutions. This eliminates the need to include the routing formulation in the model, reducing the number of constraints while increasing the number of variables. They show that only a subset of the feasible solutions need to be included in the model. Mercier et al. (2005) improve the model of Cordeau et al. (2001) by introducing the concept of restricted connections, which enable more robust solutions to the integrated routing and crew pairing problem. They compare two implementations of the Benders decomposition approach and improve the speed of convergence. Mercier and Soumis (2007) include time windows to this formulation. In a similar manner, Papadakos (2009) attempts to integrate routing and crew pairing and proposes a method based on Benders decomposition and accelerated column generation.

There have also been efforts towards solving the IFACPP. Barnhart et al. (1998c) address the daily IFACPP, but instead of simultaneously solving the two problems, they include a relaxation of the crew scheduling problem within the fleet assignment model. This relaxation does not impose pairing feasibility rules, such as maximum duties in a pairing, maximum time away from base, but covering flight legs is required. Sandhu and Klabjan (2007) also address the daily IFACPP. Their solution approaches rely on Benders decomposition and Lagrangian relaxation. While they ignore aircraft maintenance routing, they ensure plane count feasible aircraft routings. When maintenance requirements are satisfied due to the structure of the underlying network, their model fully integrates these three problems. Gao et al. (2009) develop an integrated fleet and crew robust planning method on a daily schedule. Instead of directly modeling explicit duties or pairings, they model crew connections within the integrated model. This helps them keep the model tractable. For robustness, they limit the number of fleet types and crew bases that are allowed to serve each airport.

As the above summary shows the FP and CPP have been studied widely in the literature. While there are significant contributions in terms of meth-

ods, the sheer size of the EAC problem makes most of them inapplicable. Recently, there have been more work done to address larger problems, as in Subramanian and Sherali (2008), where they address the monthly problem in a rolling horizon basis. Again the structural differences in the problems, both in terms of the cost function and the flight network, make it natural to look for alternative solution approaches. Furthermore, solving the integrated problem, IFACPP, offers greater opportunities. The existing literature on IFACPP is limited and mainly addresses the daily problem. Extending these methods to handle a monthly problem of the EAC's size is not feasible. Given the size and nature of the network and objective function we implement a method that is a combination of heuristic and exact methods. This combined effort allow us to achieve computational efficiency and improved quality in our results.

3 Problem Definition

In this section, we provide a formal definition of our problem. Our instances have two aircraft families (Airbus 320 and Boeing 737) that fly short- and medium-haul flights. The flight schedule includes approximately 27,000 flights per month to more than 100 destinations. Each aircraft family consists of several aircraft types with different capacities and operating costs. For the fleet assignment component, we aim to maximize the expected profit of assigning aircraft to flight legs. For each leg and aircraft type there is an associated expected profit, that accounts for expected revenues and cost of operating the aircraft. **We note that there are fleet assignment models in the literature that capture the impact of itinerary based demand, rather than the leg based demand model we have presented. These models are better able to capture the impact of spill and recapture of demand. Such itinerary based demand and pricing models have resulted with superior fleet assignment/capacity solutions. See for example, Barnhart et al. (2002); Wang et al. (2013) for itinerary based fleet assignment models, Atasoy et al. (2014) for an integrated scheduling, fleeting and pricing model with itinerary based demand, Sherali et al. (2009) for an integrated flight scheduling and fleet assignment model with itinerary based demands.** For the crew pairing component, we aim to minimize the costs of assigning crews to flight legs. In the integrated problem we **maximize** the total **profit** from both components of the problem.

As we address IFACPP, we make assignments for individual aircraft instead of just to an aircraft family or an aircraft subtype. This allows us to capture the CPP more accurately as connection times depend on whether the crew continue on the same aircraft or change aircraft. In doing so, we actually solve the routing problem as well, except for the maintenance requirements of the aircraft. If due to the underlying network structure, aircraft maintenance requirements are satisfied, our model also integrates the routing problem along with fleet assignment and crew pairing problems.

Next, we present the mathematical model of our integrated problem.

$$\max \sum_{a\in A}\sum_{j\in L} \Pi_{aj} x_{aj} - \sum_{p\in P} \gamma_p y_p \qquad (10)$$

s.t.

$$\sum_{a \in A} x_{aj} = 1 \quad \forall j \in L \tag{11}$$

$$\sum_{p \in P_j} y_p = 1 \quad \forall j \in L \tag{12}$$

$$\sum_{\{k \in L \cup \{t_a\} : o(k) = d(j)\}} u_{ajk} = x_{aj} \quad \forall a \in A, \forall j \in L \tag{13}$$

$$\sum_{\{k \in L \cup \{s_a\} : d(k) = o(j)\}} u_{akj} = x_{aj} \quad \forall a \in A, \forall j \in L \tag{14}$$

$$\sum_{\{j \in L : d(s_a) = o(j)\}} u_{as_aj} \leq 1 \quad \forall a \in A \tag{15}$$

$$\sum_{\{j \in L : d(s_a) = o(j)\}} u_{as_aj} = \sum_{j \in L} u_{ajt_a} \quad \forall a \in A \tag{16}$$

$$\sum_{f \in F} w_{pf} = y_p \quad \forall p \in P \tag{17}$$

$$y_p + \sum_{a \in A_f} x_{aj} \leq 1 + w_{pf}, \quad , \forall j \in L,\ \forall p \in P_j, \forall f \in F \tag{18}$$

$$x_{aj}, w_{pj}, y_p, u_{ajk} \in \{0, 1\}, \tag{19}$$

where the parameters and decision variables are defined as follows:

Sets
F: Set of aircraft types (fleet types),
P: Set of all feasible pairings
P_j: Set of pairings covering leg j actively, $P_j \subset P$
L: Set of all legs
A: Set of all aircraft
A_f: Set of aircraft of family f
L_2: $\{(j,k) : j,k \in L$ and compatible **(feasible connection time/location)**$\}$.

Parameters
Π_{aj}: Expected net profit of assigning aircraft a to leg j
γ_p: Cost of pairing p
$o(j)$: origin **airport** of flight $j \in L$
$d(j)$: destination **airport** of flight $j \in L$
s_a**: auxiliary flight that represents the start of the route flown by aircraft** $a \in A$ **(if any). The destination of this auxiliary flight is the origin airport of aircraft** $a \in A$
t_a**: auxiliary flight that represents the end of the route flown by aircraft** $a \in A$ **(if any). This auxiliary flight is compatible with all flights.**

Decision variables
x_{aj}: binary variable, 1 if aircraft a is assigned to flight j, $a \in A$, $j \in L$
u_{ajk}: binary variable, 1 if aircraft a is assigned to flight k immediately after flight j, $a \in A$, $(j,k) \in L_2$

y_p: binary variable, 1 if pairing p is chosen
w_{pf}: binary variable, 1 if pairing p is assigned to family f

The objective function (10) maximizes the total profit of IFACPP. Constraints (11) enforce that each leg is assigned to an aircraft, and constraints (12) ensure that each leg is covered actively by a pairing. Constraints (13) and (14) dictate that if $x_{aj} = 1$ then flight j has exactly one successor and one predecessor, respectively. **Constraints (15) require that there can be at most one departure from the auxiliary start flight for each aircraft. Similarly, constraints (16) require that an auircraft that has departed must end its route at the auxiliary end flight.** Constraints (17) state that each pairing is assigned to exactly one aircraft type if selected. Constraints (18) ensure that if a leg is to be covered by a pairing, then this pairing is assigned to the corresponding aircraft type. Finally, constraints (19) are the integrality constraints.

As we have 27,500 legs and 128 aircraft, the number of just the u_{ajk} binary variables exceeds 94 billion. It is possible to reduce this number by considering the compatibility of the legs, however it is not possible to reduce this number to an acceptable value. Hence, the model above only provides a mathematical explanation of our problem and it is not possible to solve this model using standard solvers or methodologies. This is our motivation in designing a time efficient solution approach to the integrated problem. In Section 4, we describe this approach in detail.

4 Solution Method

In this section, we describe our integrated approach for solving IFACPP. Our approach first employs a procedure that solves FP and CPP sequentially. Note that a very good solution for the FP phase may produce undesirable results for the CPP. Devising a mechanism that signals this information back to the FP phase, we may modify the solution in the FP phase to obtain better solutions in the CPP phase. With this in mind, we redesign our solution methodology for the FP phase to receive and process information on the subsequent CPP solution. This results **in** an iterative solution to IFACPP. Performing a number of iterations, and recording the solutions at each iteration, we apply a final selection procedure to simultaneously determine solutions for both FP and CPP, which have the best aggregate objective function value.

The real challenge in solving the FP is due to the precedence relationships, each flight assigned to an aircraft must be compatible with both the preceding and the succeeding flights. We first identify a "good" initial solution, using FP Initialization (FP_0) algorithm, that generates *chains* of flight legs with a greedy heuristic, then assigns these chains to the available aircraft to maximize the total profit of the FP phase by solving an assignment problem. The profit in the FP phase is calculated as the revenues from the ticket sales minus the operating cost of the aircraft on the given flight legs. Starting with this initial solution, we improve the solution using a mix of local search, iterated local search and exact optimization algorithms. This results in a solution that includes assignments to individual aircraft. **For the CPP we utilize an algorithm ($LNS1$), developed by Erdogan et al. (2015), which**

uses a combination of metaheuristics and exact optimization methods to solve the CPP in an iterative manner.

The procedures followed to solve the IFACPP are depicted in Figure 1. Each procedure is described in detail below.

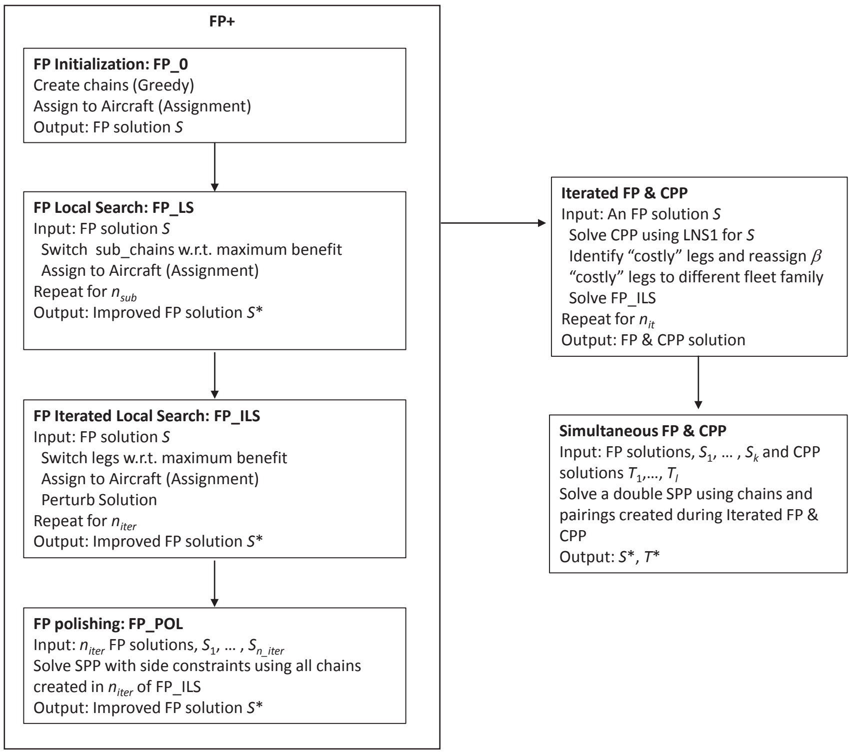


Fig. 1 Summary of solution approach

FP Initialization:
To initialize the fleet assignment procedure, we start with a time sorted leg list. Starting with the first leg, we create *chains* of flights that can be performed consecutively. We generate these chains in a greedy manner, that is, after adding the leg which arrives at time t_a to the chain, we obtain the "ready time" as $t_a + r$, where r is the minimum turn time, **which is 30 minutes**. Next, we identify the first fight leg that departs from that particular airport at or after the ready time and add that flight leg to the chain and we continue in this manner. **If there is no flight departing from that particular airport up until a certain time (14 hours for domestic and 48 hours for international locations), then there are no more feasible legs to be added to this chain** and we start a new chain from the first unassigned leg in the time sorted leg list. Once all legs are included to a chain, we concatenate feasible chains, to reduce the number of chains to the number of available aircraft. This concatenation step is different than adding legs to a chain in a greedy manner. **Here we include the possibility of repositioning of an**

aircraft and concatenate two previously generated chains if one aircraft can feasibly perform one after the other while accounting for the deadheading flight time from the arrival location of the last flight of the first chain to the departure location of the first flight of the second chain. As in the previous routine, concatenation process is also greedy hence a chain is concatenated to the earliest feasible chain. Note that the chains are not required to start and terminate at the same airport as the initial condition (locations of aircrafts, crews, etc.) at the beginning of each month might be different from one another.

Procedure FP Initialization, FP_0:

Input: A set of time sorted legs L, a set of available aircraft A

Output: A FP solution S which is a list of flight chains C^* assigned to aircraft

$L = \{l_1, l_2, l_3, ..., l_N\}$: Time sorted leg list
$A = \{a_1, a_2, a_3, ..., a_A\}$: Aircraft list
$t_d(l_i)$: departure time of leg l_i
$t_a(l_i)$: arrival time of leg l_i
r: minimum turn time
$TL(j)$: time limit for aircraft waiting time at location j, 14 hours for domestic 48 hours for international airports
$C = \emptyset$, Set of chains, initially empty
$i = 1$
For $n = 1, ..., N$
    $C[n] = L[1]$, $C = C \bigcup C[n]$
    $L = L \backslash l_i$
    Do
        Find the earliest next flight, l_k where $t_d(l_k) \geq t_a(l_i) + r$ and $o(l_k) = d(l_i)$.
        Add l_k to chain $C[n]$ so that $C[n] = \{C[n] \bigcup l_k\}$
        $L = L \backslash l_k$
        $i = k$
    While $\exists\ l_k$ where $o(l_k) = d(l_i)$ and $t_d(l_k) \leq t_a(l_i) + TL(d(l_i))$ in L.
    If $L = \emptyset$
        break;
    End If
    $n = n + 1$
End For
Do
    Initialize set of chains that can be concatenated, $\tilde{C} = C$ Identify the earliest chain, $C[i] \in \tilde{C}$
    Find the earliest chain $C[j] \in \tilde{C}$ that can be concatenated to $C[i]$, $t_d(l_k) \geq t_a(l_m) + r + \delta(o(l_m), d(l_k))$ where l_m is the last flight of $C[i]$, l_k is the first flight of $C[j]$ and $\delta(o(l_m), d(l_k))$ is the deadheading time between $o(l_m)$ and $d(l_k)$
    if ($\exists$ C[j]) concatenate $C[i]$ and $C[j]$, $C[i] = C[i] \bigcup C[j]$, $C = C \backslash C[j]$, $\tilde{C} = \tilde{C} \backslash C[j]$
    else $\tilde{C} = \tilde{C} \backslash C[i]$
While $|C| > |A|$
Assign C to A to maximize $\sum_{C[i] \in C} \sum_{j \in C[i]} \Pi_{aj}$: and obtain S.
Return S

FP local search (Sub-Chain Based):
Next, we implement a local search algorithm that updates the initial fleet assignment solution, by switching a portion of the chains created before to generate more profitable chains and reassigning to the aircraft. Given two chains, there may exist several switching points where we can cut the chains and create new chains from these switching points. Identifying all possible switching points between any pair of chains is a computationally expensive procedure, nevertheless, still doable. However due to the extensive computational time, this procedure takes too long to converge to a good solution. Therefore, instead of applying this model right after the initialization step, we perform an intermediate local search based on "sub-chains", which are sequences of flight legs that start and end at the BASE. The motivation for using these sub-chains is the fact that most of these switching points occur at the BASE due to the flight network characteristics. Therefore, we cut the long chains into sub-chains. We assume that these sub-chains are not further decomposable in the switching process. We obtain new chains by switching these sub-chains between compatible long chains. We illustrate this with a small example. Suppose there are two chains, 1-2-3-4-5-6, and a-b-c-d-e-f-g-h, with sub-chains 1-2-3, 4-5-6, a-b-c-d and e-f-g-h. **Then, the pair $(3, d)$ is a potential switching point.** Then we create new chains by exchanging these sub-chains within the long chains to create new chains. In this small example, the new chains would be 1-2-3-e-f-g-h and a-b-c-d-4-5-6. **Note that a pair of chains might have several switch points depending their compatibility.** From 128 chains created in the previous step, we generate approximately 4000 sub-chains. Hence, when we use sub-chains instead of the flight legs, we increase the granularity in the hope that the procedure converges much faster due to less number of combinations. We identify all possible switching points for all pair of chains and calculate the benefit of all possible switches using the previously assigned aircraft and sort these possible switches with respect to their marginal benefits. Next, we continue with switching the portions of the chains and reassigning these to the aircraft until we cannot find any profitable switches to perform (previously modified chains can no longer be used in the switching process). Repeating this step for a certain number iterations ($n_s ub$) quickly brings the problem to a "better" fleet assignment solution compared to the initial solution of FP_0. The details of this procedure are as follows:

Procedure: FP chains update, FP_LS
Input: An FP solution, S
Output: An improved FP solution, S^*

$S^* = S$
For $n = 1, ..., n_{sub}$
Identify the set of all possible switch points, P_{ij} between chain $C[i]$ and $C[j]$ for all i and j in solution S^*
For each switch point $p \in P_{ij}$ switch and obtain chains $\tilde{C}[i], \tilde{C}[j]$ where the first leg of $\tilde{C}[i]$ ($\tilde{C}[j]$) is the first leg of $C[i]$ ($C[j]$),
Calculate the profit , $\hat{\Pi}[p]$, of $\tilde{C}[i]$, for either new chain, assuming it is assigned to the same aircraft as $C[i]$,
Calculate the profit , $\tilde{\Pi}[p]$, of $\tilde{C}[i]$, for either new chain, assuming it is assigned to the same aircraft as $C[j]$,
Calculate benefit, $B_{ij}[p] = \max\{\hat{\Pi}[p] + \hat{\Pi}[p] - [\Pi_i + \Pi_j], \tilde{\Pi}[p] + \tilde{\Pi}[p] - [\Pi_i +$

$\Pi_j]\}$, of making this exchange at point p between chains i and j
Sort switches $p \in P_{ij}$ for all i, j, with respect to the benefit achieved in profits with the switch, $B_{ij}[p]$, in descending order.
Starting with the switch with the highest benefit, update the chains until no more profitable switches can be found. If a chain has been updated once, no further switches can be made on this chain. Record this as new FP solution S^*.
End For
Return S^*

FP iterated local search
At this point, we have a "good" solution to the FP, and we start a more detailed iterated local search. In essence, as a subroutine we use a local search algorithm that performs the switching procedure discussed above on a *leg* basis, instead of mini chains. **Using the same example, we illustrate this "leg-based switch". Recall that we have two chains, 1-2-3-4-5-6, and a-b-c-d-e-f-g-h, with sub-chains 1-2-3, 4-5-6, a-b-c-d and e-f-g-h. Then, in "mini chain-based" switch, the pair $(3, d)$ is the only potential switching point. However, assuming the compatibility of the legs the pairs $(2, b)$ and $(5, g)$ might be other potential switching points besides $(3, d)$ as now we are allowed to break mini chains. Clearly, this leg-based switch is computationally more challenging compared to the previous case as the number of potential switch points are much higher and they may occur at some airport other than the BASE.** We carry out these iterations until we reach a local maximum point where we cannot improve the solution anymore. After reaching this local maximum, we perturb the solution by randomly breaking α of the chains and creating new chains by patching these broken chains to other chains. Starting with this perturbed solution, we repeat the leg based local search algorithm **for a certain number of iterations (n_iter). The reason for implementing this perturbation is to avoid to be stuck at local optima.**

Iterating this step n_{iter} times yields an improved solution to FP.

Procedure: FP Iterated Local Search, FP_ILS
Input: An FP solution, S
Output: An improved FP solution, S^*

$S^* = S$
For $n = 1, ..., n_{iter}$
Identify all possible switch points, $p \in P_{ij}$ between all chains $C[i]$ and $C[j]$ for all i, j, in solution S^*
For each $p \in P_{ij}$, switch and obtain chains $\hat{C}[i], \hat{C}[j]$
Calculate the profit , $\hat{\Pi}[p]$, of $\hat{C}[i]$, for each new chain, assuming it is assigned to the same aircraft as $C[i]$,
Calculate the profit , $\tilde{\Pi}[p]$, of $\hat{C}[i]$, for each new chain, assuming it is assigned to the same aircraft as $C[j]$,
Calculate benefit, $B_p = \max\{\hat{\Pi}[p] + \hat{\Pi}[p] - [\Pi_i + \Pi_j], \tilde{\Pi}[p] + \tilde{\Pi}[p] - [\Pi_i + \Pi_j]\}$, of making this exchange at p between chains i and j
Sort switches $p \in P_{ij}$, for all i, j, with respect to the benefit achieved in profits with the switch, B_p, in descending order.

Starting with the switch with the highest benefit, update the chains until we cannot find any profitable switches to perform. (If a chain has been updated once, no further switches can be made on this chain). Record this as new FP solution S^*.

In solution S^*, randomly select α of the chains and randomly select one switching point in each of these chains. Force switching by artificially increasing the benefits of switching at these points.

Update S^* to this perturbed solution.

End For

Return S^*

FP polishing:

To obtain a final solution to the FP, we implement a final polishing step. In this step, we take all the chains created during the n_{iter} iterations of FP_ILS, and upon deleting duplicate chains, we solve a set partitioning problem (SPP) with side constraints so that each leg is covered once by the selected chains and each selected chain is assigned to an aircraft to maximize the profits. The details of this polishing step is as follows:

Procedure: FP polishing, FP_POL

Input: n_{iter} FP solutions, $S_1 \ldots S_{n_{iter}}$

Output: An improved FP solution, S^*

List all the chains in $S_1 \ldots S_{n_{iter}}$

Remove duplicates in the chain list to obtain $\hat{C}$

Solve **SPP_0** to obtain a solution S^*

Return S^*

SPP_0:

$$\max \sum_{c \in \hat{C}} \sum_{a \in A} \Pi_{ca} x_{ca} \tag{20}$$

subject to

$$\sum_{c \in C_j} \sum_{a \in A} x_{ca} = 1, \qquad \forall j \in L, \tag{21}$$

$$\sum_{a \in A} x_{ca} \leq 1, \qquad \forall c \in \hat{C}, \tag{22}$$

$$\sum_{a \in A, c \in \hat{C}} x_{ca} \leq N_A \tag{23}$$

$$x_{ca} \in \{0, 1\}, \quad \forall c \in \hat{C}, a \in A, \tag{24}$$

where
C_j: set of chains covering leg l, $C_j \subset \hat{C}$,
Π_{ca}: expected net profit of assigning aircraft a to chain c,
x_{ca}: binary decision variable, 1 if aircraft a is assigned to chain c, $a \in A$.
The objective function (20) requires maximizing the total expected profit by assigning aircraft to chains while the partitioning constraints (21) enforces that exactly one chain is assigned to each leg. Constraints (22) states that each aircraft is assigned at most to one chain. Constraint (23) ensures that the number of available aircraft is not exceeded.

Our overall process to determine an FP solution is summarized as follows:

Procedure FP+
Execute Procedure: FP initialization, FP_0
Execute Procedure: FP chains update, FP_LS
Execute Procedure: FP iterated local search, FP_ILS
Execute Procedure: FP polishing, FP_POL
Return S^*.

CPP:
We solve the CPP based on the fleet assignments found in FP phase. To solve the CPP, we implement the $LNS1$ algorithm of Erdogan et al. (2015) for both aircraft families. In $LNS1$, Erdogan et al. (2015) use a combination of metaheuristics and exact optimization methods to solve the CPP in an iterative manner. The procedure finds a feasible solution in a greedy manner and then randomly removes a fraction of its pairings. Thus, the solution to CPP becomes infeasible as a subset of legs now becomes uncovered. For each uncovered leg, a minimum-cost pairing is identified via a depth-first branch-and-bound algorithm. The pairing thus derived is included in the solution and the process is reiterated until a feasible solution S' to CPP is obtained. Then, the process is restarted with S' being the new working solution. After a number of iterations, as an intermediate step, an SPP is solved with the pairings obtained in **all** previous solutions to update the current working solution and the process continues iteratively from this updated solution. We refer the reader to Erdogan et al. (2015) for more details.

We solve FP and CPP sequentially using $FP+$ and $LNS1$ procedures to obtain an overall solution for the integrated problem **sequentially**. Since our FP solution gives assignments to aircraft (instead of aircraft types) constraints like connection time, which depend on whether crew changes aircraft, are correctly represented in the CPP stage. The solution obtained in this sequential manner will serve as a benchmark for evaluating our integrated solutions. Next, we describe two procedures that we developed to solve the IFACPP, the first is an iterative procedure that takes the implications of the solution of one stage to the other and iteratively updates the solution of each stage. This solution is expected to provide better results than the sequential approach as considerations from each stage is reflected in the other. The second approach addresses IFACPP through a new formulation of the problem, which can be summarized as a simplified version of the model (10)-(18) that relies on the solutions obtained during the iterative procedure. With this model we simultaneously solve the FP and CPP stages. We first describe the iterative approach.

Iterative FP & CPP :
In order to iteratively solve FP and CPP, we require a feedback mechanism from CPP phase to FP phase, indicating that some of the fleeting decisions actually lead to costly crew pairings. Although we can identify costly pairings immediately, we cannot determine the exact cause that made those pairings costly. However, one can safely assume that the flight legs covered by a pairing influence the cost of the pairing. Hence, we first identify costly pairings (deadheading plus layover costs) and then divide the pairing cost to the number of active legs in that pairing. Here, active legs, refer to the legs that the crew is not deadheading. Then, we sort all legs for both aircraft families in descending order of their costs, and reassign the most costly β legs of each family to the other family. We continue with the FP_ILS procedure after this CPP phase. We repeat this iterative process n_{it} times.

Procedure: Iterated FP and CPP
Input: An FP solution S
Output: An FP and CPP solution, S^*

For $n = 1, ..., n_{it}$
    For Fleet $f = 1, 2$
        Solve $LNS1$ using the FP solution S
        Identify the costly legs by calculating $cost_l$ = Cost of deadheading and layovers in a pairing / Number of active legs in pairing
        Sort the legs in order of descending $cost_l$
        Restrict the fleet assignment of first β legs to the other fleet family
    End For
Solve FP Iterated Local Search, FP_ILS
End For

Simultaneously solving FP & CPP :
Our motivation for analyzing the integrated problem is to obtain FP and CPP solutions that produce the best result for IFACPP. The iterative process, described above, transfers information from the CPP phase back to the FP phase in order to increase the compatibility of solution in phases, thus results with an increase in the overall profitability of the airline. Even though higher benefits are achieved

through this iterative process, we still solve FP and CPP separately, and this limits the overall improvement. To overcome this limitation, we develop a simultaneous solution procedure that handles both FP and CPP at the same time. During the iterative process, we collect several FP and CPP solutions, namely chains and pairings. In this simultaneous procedure, we develop new integer programming formulation that selects and assigns chains and pairings so that each leg is covered by exactly one chain and one pairing and make sure that those two belong to the same aircraft family. This new model can be viewed as a "double" set partitioning problem for the flight legs with additional constraints, hence is a difficult problem to solve especially considering the size of the problem. **The best known algorithm to solve this problem has an exponential complexity.** Using the same notation as in (10)–(19) and (20)–(24) for defining the sets and parameters, the new mathematical model is as follows:

$$IP : \max \sum_{a \in A} \sum_{c \in C} \Pi_{ca} x_{ca} - \sum_{p \in P} \gamma_p (y_{p1} + y_{p2}) \tag{25}$$

s.t.

$$\sum_{c \in C_j} \sum_{a \in A} x_{ca} = 1 \quad \forall j \in L \tag{26}$$

$$\sum_{c \in C} x_{ca} \leq 1 \quad \forall a \in A \tag{27}$$

$$\sum_{p \in P_j} (y_{p1} + y_{p2}) = 1 \quad \forall j \in L \tag{28}$$

$$\sum_{p \in P_j} y_{p1} - \sum_{c \in C_j} \sum_{a \in A_1} x_{ca} = 0 \quad \forall j \in L \tag{29}$$

$$\sum_{p \in P_j} y_{p2} - \sum_{c \in C_j} \sum_{a \in A_2} x_{ca} = 0 \quad \forall j \in L \tag{30}$$

$$x_{ca}, y_{pf} \in \{0, 1\}. \tag{31}$$

Decision variables
x_{ca}: binary variable, 1 if aircraft a is assigned to chain c, $a \in A$
y_{pf}: binary variable, 1 if pairing p is selected to be covered by an aircraft of family $f = \{1, 2\}$

The objective function (25) maximizes the total profit of IFACPP. Constraints (26) ensure that each leg is covered by a chain. Constraints (27) ensure that an aircraft is assigned to at most one chain. Constraints (28) guarantee that a leg is covered actively by a pairing of either aircraft families. Constraints (29) and (30) respectively ensure that if a leg is covered by a pairing of family f, then the corresponding chain should be also assigned to an aircraft that belongs to family f. In fact, Constraints (30) are redundant as long as Constraints (29) are present.

Finally, the details of the procedure are as follows:

Procedure: Simultaneous Solution
Input: k FP solutions, $S_1 \ldots S_k$, and k CPP solutions, $T_1 \ldots T_l$

Output: An overall IFACPP solution, S^*, T^*

List all the chains in $S_1 \dots S_k$
Remove duplicates in the chain list
List all the pairings in $T_1 \dots T_l$
Remove duplicates in the pairing list
Solve IP to obtain a solution S^*, T^*
Return S^*, T^*

5 Computational Study

We have tested our solution method on five instances of data sets acquired from publicly available data of EAC. We consider two aircraft families (Airbus 320 and Boing 737) that fly short- and medium-haul flights. We focus on these two aircraft families as they are frequently used for serving the same destinations due to similarity of their range and capacities. The flight schedules for the five instances consist of 26260, 26501, 27139, 27348 and finally 27,360 legs. There are 6 and 8 aircraft types (including different seating configurations and different operating costs) within the Airbus 320 and Boeing 737 families, respectively. There are a total of 128 aircraft available.

We do not have information about the initial location of aircraft and crew at the beginning of the planning period of our instances. We resolve this by adding a 1-week warm-up. This allows us to reach a feasible starting point for solving the IFACPP. We also add a 1-week cool-down period at the end of the month. As we include 1 week of warm-up in the beginning and 1 week of cool-down period at the end of each month, we solve a 42-day problem instead of a one-month problem for each instance, hence these numbers correspond to a 6-week planning horizon. When reporting results, we only include results related to the flights within the particular month. **Other than the change in number of legs in the instances, based on the month, we see that some new destinations are added and some others see significant changes in the number of flights schedules to and from those airports. The increase in the number of flights to/from an airport can be as high as 70% as the season changes. The average increase in the number of flights to/from an airport with changed number of total flights is 31%.**

In the FP phase, for profit and cost parameters, we use representative figures that reflect true values in terms of proportions and magnitudes. **For aircraft operating costs we relied on available industry data for different aircraft types, although these costs may not be exact costs incurred by EAC they would be representative in terms of order of magnitude and relative proportions with each other.** Similarly, in the CPP phase, representative values are used for deadheadings as well as layover costs. The number of deadheadings and layovers are based on the number of people in the crew, e.g. a layover of a crew of two people counts as two layovers. All the computational experiments are carried out on a 64-bit Windows Server with two 2.4 GHz Intel Xeon CPU's and 24 GB RAM. The algorithms are implemented using C++, and CPLEX Concert Technology.

As the computational study is carried out on the publicly available data of EAC, we do not have a real-life benchmark to compare our solutions with. To serve as a basis of comparison, we first generate solutions using our methods to solve FP and CPP sequentially. These serve as benchmarks in evaluating our integrated solutions. We also compare our results from different integrated methods in terms of solution quality and computational time.

In Table 1, we present the benchmark results, which is obtained by solving the FP and CPP phases sequentially for instances 1 through 5. The column "FP" presents the profits of the FP Phase (obtained by solving FP+). In reporting these profits, we deduct the cost of covering each leg with the least profitable aircraft, as this is a fixed cost and does not depend on the solution. "CPP" presents the costs of the CPP Phase (obtained by solving LNS1) and "Profit" presents the difference between these two values. Finally, the column "CPU Time" presents solution times. In our experimentation, in the FP stage, we chose the iteration limits, $n_{sub} = 200$, $n_{iter} = 200$. In the CPP stage, we have set $k_{max} = 100$, $s_{max} = 25$, and $\alpha = 0.7$ for LNS1. For the details of these parameters, we refer the reader to Erdogan et al. (2015). Here it is worth noting the big scale difference between FP and CPP. The costs of deadheading and layovers constitute approximately 1% of the profits in the FP. This is due to the fact that crew related costs, like salaries, are not reflected in the objective function of CPP, due to the nature of the problem in EAC.

Table 1 The profits in the FP phase (in US$), the operating costs in the CPP phase (in US$), and the corresponding CPU times (in seconds) of the sequential solution approach.

Instance	FP	CPP	Profit	CPU
1	217,792,248	2,337,545	215,454,703	6154
2	219,062,134	2,221,802	216,840,332	5785
3	222,951,575	3,013,809	219,937,766	6104
4	220,767,739	2,491,554	218,276,185	6814
5	223,041,503	2,363,896	220,677,607	9704

Table 2 presents the results obtained by iteratively solving the FP and CPP for instances 1 through 5. Here the number of iterations, n_{it}, is set to 10, and the number of legs switched to other family, β, is 30. The columns are similar to those of Table 1 except that we add two columns showing the dollar change in "Profit" and "CPP Cost", these are denoted by "$\triangle$ Profit" and "$\triangle$ CPP Cost". Note that negative changes in CPP cost reflect better CPP solutions. By the construction of the mechanism, the solution of the FP phase cannot improve in the iterative solution; we only impose extra restrictions (such as a flight leg cannot be assigned to a particular family), which in turn decrease the profits of the FP phase. However, we also expect the operating costs of the CPP Phase to improve and expect that the improvement will be high enough to compensate the losses in the FP phase with respect to the benchmark solution, and hence the overall profits will be higher. As we observe from Table 2, this is the case for instances 3 and 4. In these two instances, the overall profits increase with a significant reduction of CPP costs over the benchmark solution. Indeed, in all instances there is a significant decrease in CPP costs compared to the CPP costs in the sequential approach, an average of $325,981, corresponding to a 12% decrease, with the highest decrease being 25%. However in some instances, the overall profits are lower than our benchmark solutions. In these instances the savings from CPP phase were not sufficient to offset the loses in FP phases. The CPU times are presented in the last column. Even though these values are approximately 10 times that of the sequential solution, given that the planning period of the IFACPP is one month, they are still acceptable.

Table 2 The results, the improvements over the benchmark (in US$) and the CPU times (in seconds) of the iterative solution of FP and CPP.

Instance	FP	CPP	Profit	△ Profit	△ CPP Cost	CPU
1	217,018,956	2,064,553	214,954,403	-500,300	-272,992	62713
2	218,722,460	2,162,463	216,559,997	-280,335	-59,339	60607
3	222,542,598	2,245,978	220,296,620	358,854	-767,831	64210
4	220,491,978	2,111,114	218,380,864	104,679	-380,440	70275
5	222,883,954	2,214,592	220,669,362	-8,245	-149,304	98152

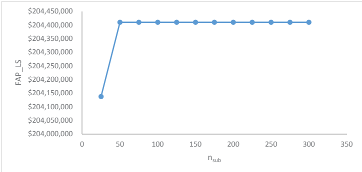
Finally, Table 3 presents the results obtained by simultaneously solving the FP and CPP for instances 1 through 5. As can be observed from Table 3, with the simultaneous solution approach for IFACPP, we obtain better results compared to the benchmark solution for all five instances. The average improvement of the CPP Cost is around $263,271, corresponding to a 10% decrease in cost, with the highest decrease being 27.45%. The average overall improvements in profits is 2.4 millions. These results prove that solving FP and CPP in an integrated manner can generate significant benefits over the sequential solution approach. We claim that the results will be more significant for airlines where CPP costs also include crew salaries.

Table 3 The results, the improvements over the benchmark (in US$) and the CPU times (in seconds) of the simultaneous solution of FP and CPP.

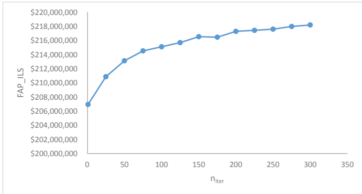
Instance	FP	CPP	Profit	△ Profit	△ CPP Cost	CPU
1	217,792,248	2,313,000	215,479,248	2,337,545	-24,545	743
2	219,062,134	2,188,300	216,873,834	2,221,802	-33,502	832
3	222,542,598	2,186,600	220,355,998	2,604,832	-827,209	1160
4	220,707,695	2,233,300	218,474,395	2,431,510	-258,254	979
5	222,883,954	2,191,051	220,692,903	2,206,347	-172,845	864

In order to assess the sensitivity of the algorithms to the algorithm parameters we ran a sensitivity analyses on n_{sub} in the FP chains update procedure, n_{iter} in the FP Iterated Local Search procedure, n_{it} and β in the Iterated FP and CPP procedure. For all parameters it was seen that the results converged for large enough iterations. Indeed for $n_{sub} > 50$ the result of procedure FAP_LS stayed constant. For $n_{iter} > 50$ the change in FP_ILS was less than 0.3%. For n_{it} the change in profit was less than 0.25% and for β the change in profit was less than 0.15%. The progress of the algorithms with respect to different parameter values can be seen in Figure 2 below.

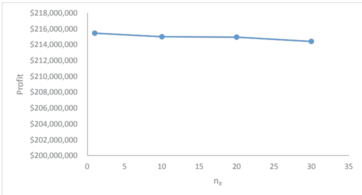
Finally, to better gauge the quality of our solution we compared the profit we obtain for the IFACPP via our solution approach with an upper bound. We implemented an upper bound using a multi-commodity network flow problem formulation for the Fleet Assignment component and ignored the crew pairing cost. We solved an LP relaxation of the multi-commodity network flow problem. Based on this the average deviations from the upper



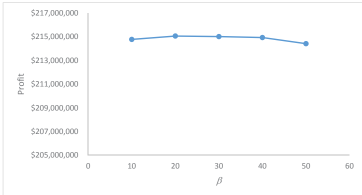
(a) FAP_LS results w.r.t. n_{sub}



(b) FAP_ILS results w.r.t. n_{iter}



(c) Iterated FP and CPP results w.r.t. n_{it}



(d) Iterated FP and CPP results w.r.t. β

Fig. 2 Sensitivity of results with respect to algorithm parameters

bound for the 5 instances is 8.1%. The deviations from the upper bound is as follows for the 5 instances:

Table 4 Deviation from upper bound.

Instance	Deviation from UB
1	7.89%
2	7.98%
3	8.10%
4	8.62%
5	8.15%

6 Conclusion

In this paper, we study the integrated fleet assignment and crew pairing problem and propose both iterative and simultaneous solution approaches to improve the profitability of the airline compared to the traditional sequential approach. Our model completely integrates the FP and CPP, but neglects aircraft maintenance constraints of aircraft routing component. In the case where aircraft maintenance requirements are satisfied due to the structure of the underlying flight network, our model also integrates the routing problem. Indeed, in the addressed case, routine maintenance checks are achieved during the night when a large portion of aircraft are grounded in the Base.

The particular variant of the IFACPP we study in this paper arises in many airlines, in particular in airlines where crews are paid based on a fixed salary. The aperiodicity of the flight schedule, a feature common in many European airlines, requires addressing the monthly problem, which makes the problem size significantly big. We show that by taking into account the network structure, in this case a single base with many point to point flights, we are able to generate good solutions in reasonable time.

Our major contribution is the development of a computationally efficient solution to the monthly integrated problem. We solve the monthly problem with approximately 27,000 flight legs and over 100 destinations using heuristic and exact methods in combination. We perform a computational study on monthly flight schedules of a European Airline using publicly available data and representative revenue and cost figures. Based on the results on five different instances, we observe that our algorithm provides solutions with significant cost savings in CPP Phase and improve the overall profits compared to the sequential approach.

References

J. Abara. Applying integer linear programming to the fleet assignment problem. *Interfaces*, 19(4):20–28, 1989.

R. K. Ahuja, J Goodstein, A. Mukherjee, J B. Orlin, and D. Sharma. A very large-scale neighborhood search algorithm for the combined through-fleet-assignment model. *INFORMS J. on Computing*, 19(3):416–428, July 2007.

R. Anbil, E. Gelman, B. Patty, and R. Tanga. Recent advances in crew-pairing optimization at american airlines. *Interfaces*, 21:62–74, 1991.

J. P. Arabeyre, J. Fearnley, F. C. Steiger, and W. Teather. The airline crew scheduling problem: A survey. *Transportation Science*, 3(2):140–163, 1969.

B. Atasoy, M. Salani, and M. Bierlaire. An integrated airline scheduling, fleeting, and pricing model for a monopolized market. *Computer-Aided Civil & Infrastructure Engineering*, 29(2):76 – 90, 2014. ISSN 10939687.

A. Aydemir-Karadag, B. Dengiz, and A. Bolat. Crew pairing optimization based on hybrid approaches. *Computers & Industrial Engineering*, 65:87 – 96, 2013.

A. Azadeh, G. Asadipour, H. Eivazy, and S. Nazari-Shirkouhi. A unique hybrid particle swarm optimisation algorithm for simulation and improvement of crew scheduling problem. *International Journal of Operational Research*, 13(4):406 – 422, 2012.

A. Azadeh, M. H. Farahani, H. Eivazy, S. Nazari-Shirkouhi, and G. Asadipour. A hybrid meta-heuristic algorithm for optimization of crew scheduling. *Applied Soft Computing*, 13:158 – 164, 2013.

C. Barnhart, E. Johnson, R. Anbil, and L. Hatay. A column generation technique for the long-haul crew assignment problem. In T. Ciriano and R. Leachman, editors, *Optimization in Industry: Volume "II"*. John Wiley and Sons, U.K., 7–22 edition, 1994.

C. Barnhart, N. L. Boland, L. W. Clarke, E.L. Johnson, G. L. Nemhauser, and R. G. Shenoi. Flight string models for aircraft fleeting and routing. *Transportation Science*, 32(3):208–220, 1998a.

C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998b.

C. Barnhart, F. Lu, and R. Shenoi. Integrated airline schedule planning. In G. Yu, editor, *Operations Research in the Airline Industry*, volume 9 of *International Series in Operations Research and Management Science*, pages 384–403. Springer US, 1998c.

C. Barnhart, T.S. Kniker, and M. Lohatepanont. Itinerary-based airline fleet assignment. *Transportation Science*, 36(2):199–217, 2002.

C. Barnhart, P. Belobaba, and A. R. Odoni. Applications of operations research in the air transport industry. *Transportation Science*, 37(4):368–391, 2003a.

C. Barnhart, A. M. Cohn, E. J. Johnson, D. Klabjan, G. L. Nemhauser, and P. H. Vance. Airline crew scheduling. In R. W. Hall, editor, *Handbook of Transportation Science*. Kluwer Academic Publishers, Norwell, 2 edition, 2003b.

N. Bélanger, G. Desaulniers, F. Soumis, J. Desrosiers, and J. Lavigne. Weekly airline fleet assignment with homogeneity. *Transportation Research Part B: Methodological*, 40(4):306 – 318, 2006.

M. E. Berge and C. A. Hopperstad. Demand driven dispatch: A method for dynamic aircraft capacity assignment, models and algorithms. *Operations Research*, 41(1):153–168, 1993.

E.R. Butchers, P.R. Day, A.P. Goldie, S. Miller, J.A. Meyer, D.M. Ryan, A.C. Scott, and C.A. Wallace. Optimized crew scheduling at Air New Zealand. *Interfaces*, 31(1):30–56, 2001.

L. Cavique, C. Rego, and I. Themido. Subgraph ejection chains and tabu search for the crew scheduling problem. *Journal of the Operational Research Society*, 50 (6):608 – 616, 1999.

C-H Chen, T-K Liu, and J-H Chou. Integrated short-haul airline crew scheduling using multiobjective optimization genetic algorithms. *IEEE Transactions On Systems, Man, and Cybernetics: Systems*, 43(5):1077 – 1090, 2013.

L. W. Clarke, C. A. Hane, E. L. Johnson, and G. L. Nemhauser. Maintenance and crew considerations in fleet assignment. *Transportation Science*, 30(3):249–260, 1996.

A. M. Cohn and C. Barnhart. Improving crew scheduling by incorporating key maintenance routing decisions. *Operations Research*, 51(3):387–396, 2003.

J. Cordeau, S. Goran, F. Soumis, and J. Desrosiers. Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science*, 35 (4):375–388, 2001.

M. S. Daskin and N. D. Panayotopoulos. A lagrangian relaxation approach to assigning aircraft to routes in hub and spoke networks. *Transportation Science*, 23(2):91–99, 1989.

Guang-Feng Deng and Woo-Tsong Lin. Ant colony optimization-based algorithm for airline crew scheduling problem. *Expert Systems with Applications*, 38(5):5787 – 5793, 2011.

G. Desaulniers, J. Desrosiers, Y. Dumas, M. M. Solomon, and F. Soumis. Daily aircraft routing and scheduling. *Management Science*, 43(6):841–855, 1997.

J. Desrosiers M. Gamache F. Soumis Desaulniers, G. Crew scheduling in air transportation. In T. Crainic and G. Laporte, editors, *Fleet Management and Logistics*. Kluwer Academic Publishers, Norwell, 1998.

V. Dück, F. Wesselmann, and L. Suhl. Implementing a branch and price and cut method for the airline crew pairing optimization problem. *Public Transport*, 3: 43–64, 2011.

T. Emden-Weinert and M. Proksch. Best practice simulated annealing for the airline crew scheduling problem. *Journal of Heuristics*, 5:419–436, 1999.

G. Erdogan, M. Haouari, M Ormeci Matoglu, and O.O. Ozener. Solving a large-scale crew pairing problem. *Journal of Operational Research Society*, 66:1742–1754, 2015.

C. Gao, E. Johnson, and B.Smith. Integrated airline fleet and crew robust planning. *Transportation Science*, 43(1):2–16, 2009.

I. Gershkoff. Optimizing Flight Crew Schedules. *Interfaces*, 19:29–43, 1989.

B. Gopalakrishnan and E. Johnson. Airline crew scheduling: State-of-the-art. *Annals of Operations Research*, 140:305–337, 2005.

Z. Gu, E. L. Johnson, G. L. Nemhauser, and W. Yinhua. Some properties of the fleet assignment problem. *Operations Research Letters*, 15(2):59 – 71, 1994.

C.A. Hane, C. Barnhart, E.L. Johnson, R.E. Marsten, G.L. Nemhauser, and G. Sigismondi. The fleet assignment problem: Solving a large-scale integer program. *Mathematical Programming*, 70(1-3):211–232, 1995.

M. Haouari, H.D. Sherali, F.Z.l Mansour, and N. Aissaoui. Exact approaches for integrated aircraft fleeting and routing at Tunisair. *Computational Optimization and Applications*, 49(2):213–239, 2011.

A. Kasirzadeh, M. Saddoune, and F. Soumis. Airline crew scheduling: models, algorithms, and data sets. *EURO Journal on Transportation and Logistics*, pages 1–27, 2015. ISSN 2192-4376.

D. Klabjan, E. L. Johnson, G. L. Nemhauser, E. Gelman, and S. Ramaswamy. Airline crew scheduling with time windows and plane-count constraints. *Transportation Science*, 36(3):337–348, 2002.

D. Levine. Application of a hybrid genetic algorithm to airline crew scheduling. *Computers & Operations Research*, 23(6):547 – 558, 1996.

M. Lohatepanont and C. Barnhart. Airline schedule planning: Integrated models and algorithms for schedule design and fleet assignment. *Transportation Science*, 38(1):19 – 32, 2004. ISSN 00411655.

A. Mercier and F. Soumis. An integrated aircraft routing, crew scheduling and flight retiming model. *Comput. Oper. Res.*, 34(8):2251–2265, August 2007. ISSN 0305-0548.

A. Mercier, J. Cordeau, and F. Soumisa. A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. *Comput. Oper. Res.*, 32(6):1451–1476, June 2005. ISSN 0305-0548.

H.T. Ozdemir and C. K. Mohan. Flight graph based genetic algorithm for crew scheduling in airlines. *Information Sciences*, 133(34):165 – 173, 2001.

G. Paleologo and J.L. Snowdon. Airline optimization. In A . Ravi Ravindran, editor, *Operations Research and Management Science Handbook*, chapter 20, pages 1–31. CRC Press, 2 edition, 2007.

A. Panayiotis, P. Sanders, T. Takkula, and D. Wedelin. Parallel integer optimization for crew scheduling. *Annals of Operations Research*, 99(1-4):141 – 166, 2000.

N. Papadakos. Integrated airline scheduling. *Comput. Oper. Res.*, 36(1):176–195, January 2009. ISSN 0305-0548.

B. Rexing, C. Barnhart, T. Kniker, A. Jarrah, and N. Krishnamurthy. Airline fleet assignment with time windows. *Transportation Science*, 34(1):1–20, 2000.

R. A. Rushmeier and S. A. Kontogiorgis. Advances in the optimization of airline fleet assignment. *Transportation Science*, 31(2):159–169, 1997.

D. M. Ryan. The solution of massive generalized set partitioning problems in aircrew rostering. *The Journal of the Operational Research Society*.

M. Saddoune, G. Desaulniers, and F. Soumis. Aircrew pairings with possible repetitions of the same flight number. *Computers & Operations Research*, 40(3): 805 – 814, 2013.

R. Sandhu and D. Klabjan. Integrated airline fleeting and crew-pairing decisions. *Operations Research*, 55(3):439–456, 2007.

H. D. Sherali, E. K. Bish, and X. Zhu. Airline fleet assignment concepts, models, and algorithms. *European Journal of Operational Research*, 172(1):1 – 30, 2006.

H.D. Sherali, K. Bae, and M. Haouari. Integrated airline schedule design and fleet assignment: Polyhedral analysis and Benders' decomposition approach. *INFORMS Journal on Computing*, 2009.

H.D. Sherali, K. Bae, and M. Haouari. A benders decomposition approach for an integrated airline schedule design and fleet assignment problem with flight retiming, schedule balance, and demand recapture. *Annals of Operations Research*, 210(1):213–244, 2013. ISSN 0254-5330.

R.A Subramanian, R. P. Scheff, J. D. Quillinan, D. S. Wiper, and R. E. Marsten. Coldstart: Fleet assignment at Delta Air Lines. *Interfaces*, 24(1):104–120, 1994.

S. Subramanian and H.D. Sherali. An effective deflected subgradient optimization scheme for implementing column generation for large-scale airline crew scheduling problems. *INFORMS Journal on Computing*, 20(4):565 – 578, 2008. ISSN 10919856.

P.H. Vance, C. Barnhart, E.L.Johnson, and G.L. Nemhauser. Airline crew scheduling: A new formulation and decomposition algorithm. *Operations Research*, 45: 183–200, 1997.

D. Wang, S. Shebalov, and D. Klabjan. Attractiveness-based airline network models with embedded spill and recapture. *Journal of Airline and Airport Management*, 4(1):1 – 25, 2013.